1980-06

Parametric simulation of infantry tactics and equipment (DISMOUNTED-STAR)

Carpenter, Howard John; Thurman, Edward Eugene

http://hdl.handle.net/10945/26151

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PARAMETRIC SIMULATION OF INFANTRY TACTICS
AND EQUIPMENT
(DISMOUNTED STAR)

by

Howard John Carpenter

and

Edward Eugene Thurman

June 1980

Thesis Co-Advisors: E.P. Kelleher
S.H. Parry

Approved for public release; distribution unlimited

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Parametric Simulation of Infantry Tactics and Equipment (DISMOUNTED-STAR)	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1980
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Howard John Carpenter Edward E. Thurman	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	12. REPORT DATE June 1980
	13. NUMBER OF PAGES 270
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Infantry
STAR
Army
Combat Simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis presents a stochastic simulation model of dismounted infantry combat that is designed to be used in conjunction with the Simulation of Tactical Alternative Responses (STAR) combat model. The tactics modelled, the assumptions made, and the interface requirements are detailed, with the computer code that is used to execute the model included. An overview of the basic STAR Ground Model

DD FORM 1 JAN 73 1473 (Page 1) EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

is discussed in moderate detail to provide the reader a foundation on which to place the discussion of the dismounted model. The input requirements to use the model are explained so that this thesis can become an initial user's manual in the use of the model. The definitions of the purpose of the routines and events, the global variables, entities and sets are provided to assist the reader attempting to understand the STAR model.

Parametric Simulation of Infantry Tactics and Equipment
(DISMOUNTED-STAR)

by

Howard John Carpenter
Captain, United States Army
BS, United States Military Academy

and

Edward Eugene Thurman
Captain, United States Army
BA, University of Washington

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL

June 1980

ABSTRACT

This thesis presents a stochastic simulation model of dismounted infantry combat that is designed to be used in conjunction with the Simulation of Tactical Alternative Responses (STAR) combat model. The tactics modelled, the assumptions made, and the interface requirements are detailed, with the computer code that is used to execute the model included. An overview of the basic STAR Ground Model is discussed in moderate detail to provide the reader a foundation on which to place the discussion of the dismounted model. The input requirements to use the model are explained so that this thesis can become an initial user's manual in the use of the model. The definitions of the purpose of the routines and events, the global variables, entities and sets are provided to assist the reader attempting to understand the STAR model.

ACKNOWLEDGMENT

This is to acknowledge the invaluable assistance provided by Professor Sam H. Parry and Lieutenant Colonel Edward P. Kelleher Jr. in the development of the dismounted model. Additionally, appreciation is expressed to Professor James K. Hartman for time and guidance provided in the development of the model. Finally, we wish to acknowledge the assistance of MS Jody Shirley and Captain Jim Allen of the Directorate of Combat Developments, the United States Army Infantry School, Fort Benning, Georgia. Without their assistance in locating data and modelling methodologies, the development of the model would have been greatly delayed.

TABLE OF CONTENTS

LIST OF TABLES

Table No.

LIST OF FIGURES

Figure No.

I. INTRODUCTION

The development of the most effective armed force to meet the challenges of the modern battlefield is a complex problem of the military decision maker of today. Changes in technology produce new supporting systems on an ever changing basis. Decision makers are forced to operate within a most stringent budget which makes it imperative that the best systems attainable be introduced into the inventory.

The combined arms simulation model has in recent years been one of the most widely used tools for the evaluation of proposed and existing weapon systems. For this reason the Simulation of Tactical Alternative Responses (STAR) combined arms model was developed by Wallace and Hagewood Ref.1 . This thesis is an enhancement of STAR incorporating both Blue and Red dismounted infantry forces within the battalion level combined arms model.

Several existing models which are capable of modelling dismounted infantry forces are discussed in Chapter II.

Chapter III introduces the reader to STAR and presents a brief discussion of the use of SIMSCRIPT in STAR.

Chapter IV is a general discussion of the methods used to model infantry soldiers. Topics discussed include the soldier's physical representation, target selection, detection, firing, casualty assessment and movement.

Chapter V discusses some of the areas related to the infantry portion of the model which are now being developed for future inclusion in the model.

Appendix A is a paper by Lieutenant Colonel Edward P. Kelleher Jr., which discusses the original ground model as it exists without dismounted infantry play.

Appendix B covers the same topics discussed in Chapter IV. However this section presents a greater amount of detail to the reader who is familiar with the methods and mathematics of combat models.

Appendix C reviews all routines and events modified in the ground model to allow for the play of dismounted infantry.

Appendix D is a listing of new routines and events introduced to the STAR model. Included in this section is an explanation of the variables used in each routine, a listing of the computer code and a line by line explanation of what the code does.

Appendix E discusses the structure and content of the lethality, accuracy, and casualty inputs for the infantry weapons and entities.

Appendix F provides a listing of the definitions of global variables, entity attributes, events, routines, and sets used in the STAR model.

II. STATE OF THE ART

The purpose of this chapter is to present the reader an overview of two previously developed models which model dismounted infantry. The two models are CARMONETTE and ASARS. The discussion of each model covers its origin, purpose of development, entity representation, terrain, movement, fire, detection and casualty assessment.

The discussion of CARMONETTE which follows is derived from the Army Concepts Analysis Agency report: CARMONETTE, Volumes I-III, prepared by General Research Corporation in November 1974. CARMONETTE is a Monte Carlo, event sequenced simulation of ground combat. The original model was developed by the Research Analysis Corporation in the late 1950's. The principal activities represented in the model are movement, target detection, firing and communications. The resolution of CARMONETTE is variable. The basic element represented is referred to as a unit. The unit, as defined in CARMONETTE, is not associated with a standard military organization. The composition of a unit can range from a single combatant to a platoon of combatants. Combatants may represent either soldiers or vehicles. Units may contain both kinds of combatants. The model is capable of representing not more than forty-eight units on each side of the battle. When the unit is defined to represent an individual soldier, CARMONETTE is simulating platoon combat.

The simulation of battalion combat is achieved by defining the unit as a platoon. It is significant that all components of the unit must be located in the same grid square at all times. There is therefore, a direct relationship between unit size and grid size.

Terrain representation in CARMONETTE is characterized by a battlefield divided into grid squares. The maximum size of the battlefield is sixty by sixty-three grid squares. The size of a grid square is selected by the user to correspond with the size of the unit. Table 2-1 further depicts the relationship between grid size and unit size. Note that in Table 2-1, a unit representing a platoon calls for a grid square of size 250 meters. This results in a battlefield of size fifteen by sixteen kilometers. If a unit represents the individual soldier and the user selects a grid size of 10 meters, the size of the battlefield is six hundred by six hundred thirty meters.

The terrain is digitized and described explicitly in terms of:

a. Elevation

b. Height of vegetation

c. Trafficability of roads

d. Cross-country trafficability

e. Cover

f. Concealment

Each grid square has an average elevation that is used in

RELATION BETWEEN GRID SIZE, UNIT SIZE, FORCE SIZE, AND ZONE OF ACTION

GRID SIZE (m)	APPROXIMATE UNIT SIZE				MAXIMUM FORCE SIZE		MAXIMUM ZONE OF ACTION	
	INFANTRY	MECH INFANTRY	ARTILLERY	AVIATION AIRCRAFT	INFANTRY	MECH INFANTRY	WIDTH	DEPTH
10	1 man	NA	1 tube	NA	2 PLTS	NA	600 m	630 m
25	2 men	1 veh	2 tubes	1	1 CO	1 CO	1500 m	1575 m
50	½ SQD	2 veh	4 tubes	1	1 BN	1 BN	3000 m	3150 m
100	1 SQD	3 veh	6 tubes	2	2 BNs	2 BNs	6000 m	6300 m
250	1 PLT	7 veh	12 tubes	4	4 BNs	4 BNs	15000 m	15750 m

Table 2-1

determining slopes and lines of sight between grid squares. The average height of vegetation is added to the elevation of the intervening terrain to determine intervisibility. Modelling terrain using this method results in a terrain model like that in Figure 2-1.

Movement in CARMONETTE is based on the premise that each element represented by the unit is located within the same grid square, and the center of the unit is at the center of the grid square. Therefore, movement is from grid square center to grid square center. When there is a requirement to move, the center of a new grid square is selected as the destination. The route selected to the destination is the most direct path that moves from grid center to grid center. An example of this movement is presented in Figure 2-2.

The speed at which a unit moves depends upon the slope between adjacent grids, the trafficability index of the terrain, and a mobility index assigned to the unit. Detection in CARMONETTE is modeled at the unit level. Therefore, when one member of another unit (for example a platoon) detects any member of another unit each member of the detecting unit detects all members of the detected unit. When a firing event takes place all members of the target unit are subject to and vulnerable to the fire. The probability of hit is determined for each member, Similarly, the probability of kill given a hit is individually

Comparison of CARMONETTE Terrain

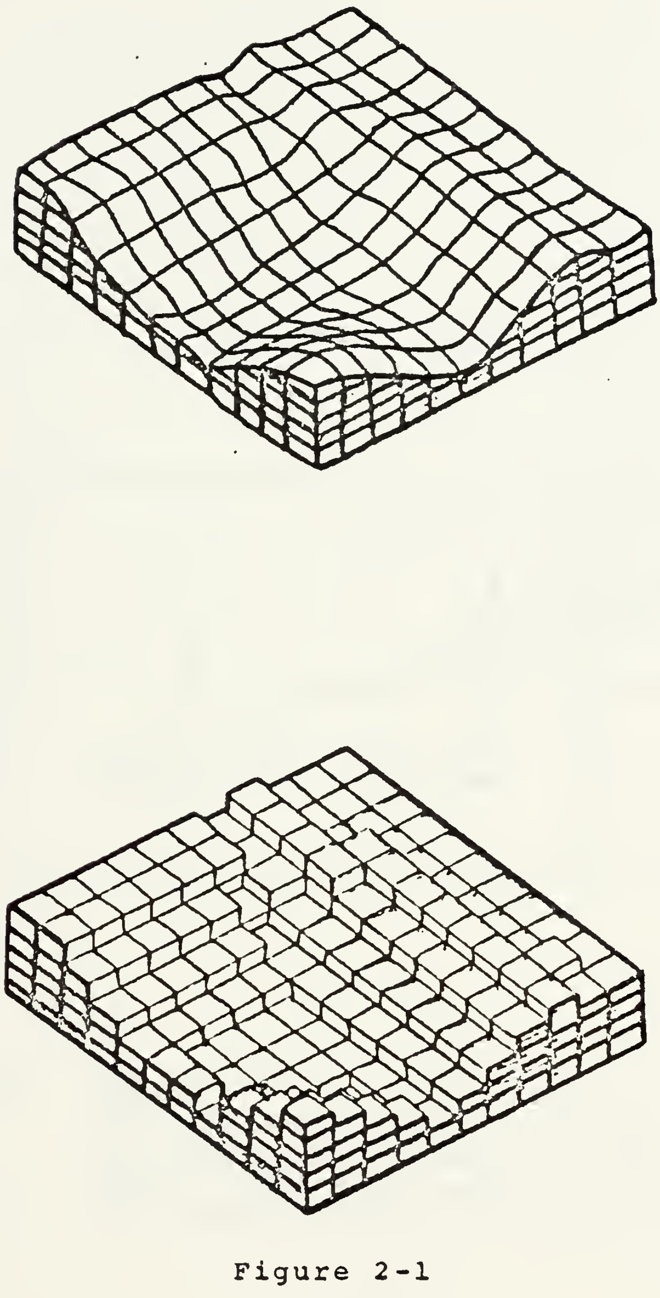
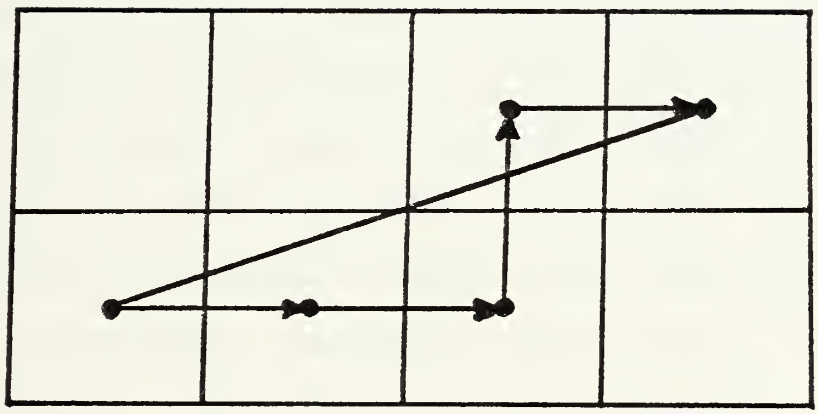


Figure 2-1

Note: Real terrain as depicted in the upper portion of this figure (with a grid overlayed) is modelled in CARMONETTE as shown in the lower figure. Each grid square is assigned an average elevation which is then used to represent the elevation of the entire grid square.

MOVEMENT IN CARMONETTE



NOTE: The route from a current position to a destination point is the most direct path that moves from grid square to adjacent grid square center. There are four choices of direction at each step.

Figure 2-2

determined.

The ARMY SMALL ARMS REQUIREMENT STUDY BATTLE MODEL (ASARS Battle Model) originated as an outgrowth of the Small Arms Weapons Study (SAWS) initiated in 1965. The version of the model discussed in this paper is based on documentation from the United States Army Combat Developments Command Infantry Agency, entitled ASARS II, Narrative Summary, dated April 1972. The ASARS Battle Model is a two sided high resolution model of dismounted infantry. The model was initially developed with the purpose of investigating weapons performance characteristics, ballistic dispersions, aim error, area coverage, rate of fire and lethality.

The resolution of the model is to the individual infantry soldier. Fireteams, squads, and the platoon are identified for maneuver purposes. Model design restricts battle simulation size to the infantry platoon versus infantry squad level, or smaller,

The terrain in this model is also digitized and is similar to the DYNTACS terrain model. The grid size ranges from 6 meters to 100 meters on a side. The usual grid size is 6.25 meters by 6.25 meters. Unlike CARMONETTE, the elevation in any grid square varies. Elevations are determined by linear interpolation along a line drawn between two opposite vertices of the grid square. The top of a grid square can be viewed as the intersection of two triangular planes. The vertex pair between which the

diagonal is drawn is randomly selected prior to the simulation. Figure 2-3 presents several examples of grid squares which result from this methodology.

Movement in ASARS is modelled in the same manner as in the DYNTACS model. Routes are generally specified by the user through the use of a series of Movement Control Points. Units may, however, deviate from the preselected route in response to the tactical situation. Factors such as minefields, suppressive direct fires, and artillery influence the individual's movement decision. Figure 2-4a presents an example of a preplanned route. A typical terrain profile of an element's vertical path as it moves along the route can be generated (Figure 2-4b).

At each point on the terrain where the route crosses the boundary of one of the triangular planes a slope determination is made and used to adjust the rate of movement of elements. The points where a route crosses a boundary are referred to as Plane Departure Points. Elevations are also calculated at these points in the determination of line of sight between two elements.

Detections in the ASARS model are at the individual element level. This means that detections are all on a one on one basis. When a determination to fire is made, each round is tracked individually to the target. If the round hits the target, an assessment is made to determine which body part was hit and whether the target was incapacitated

ASARS TERRAIN REPRESENTATION

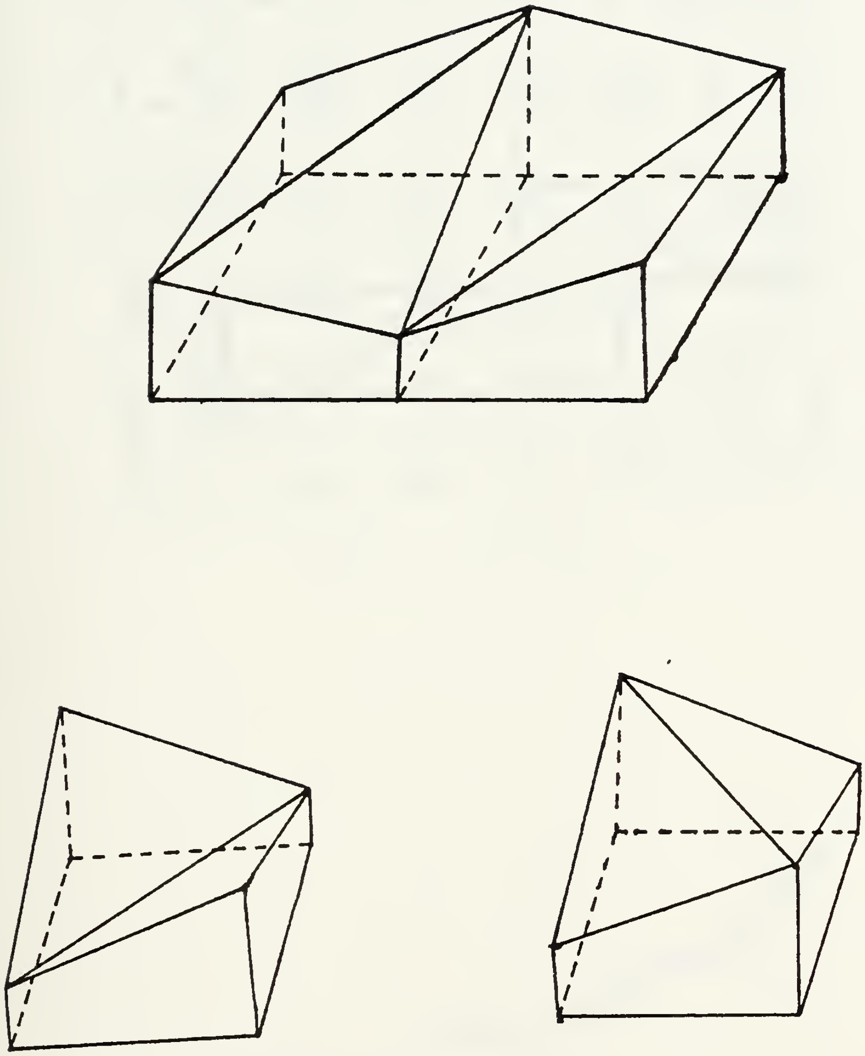


Figure 2-3

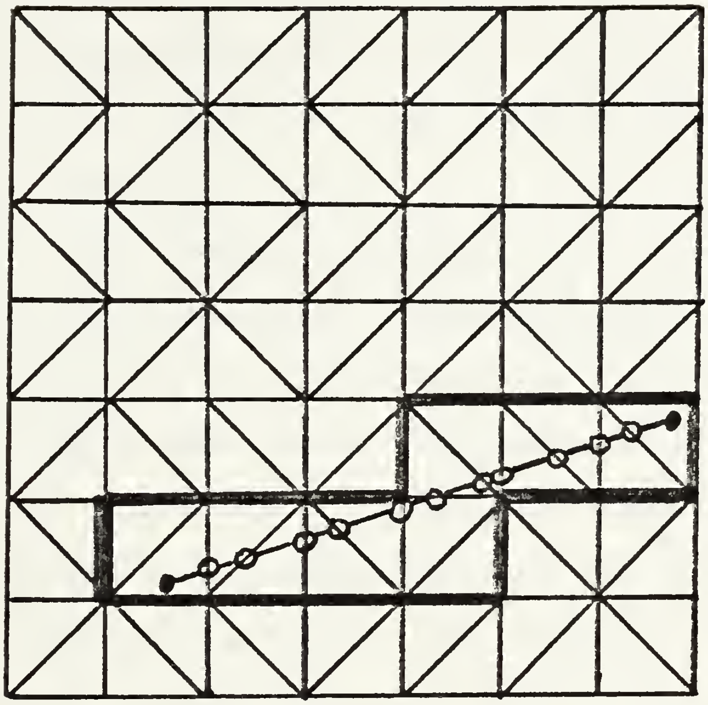


Figure 2-4a Top View

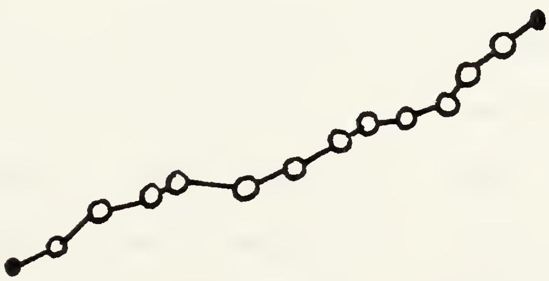


Figure 2-4b Terrain Profile

by the round. The fire and casualty assessment play in ASARS is a reflection of the purpose for which the model was designed, to assess the effectiveness of small arms. This requires the use of extensive and accurate weapons accuracy and lethality routines.

Both of the models discussed are successful in fulfilling the requirements of their original design. ASARS is most successful in capturing the actions of the individual soldier. However the resolution is of such a high level that the modelling of combat is restricted to the platoon and squad level. CARMONETTE, on the other hand, is capable of modelling battalion level combat; but this is accomplished only through the sacrifice of resolution at the individual soldier level. This resolution is lost because the detection, target selection, and firing actions are centralized at the unit level. For example, if one element of a unit in CARMONETTE detects an element of an opposing unit, all elements of the detecting unit also make the detection. Resolution is also lost with the requirement that all members of a unit be contained in a given grid square at any point in time. Additionally, the exact (x,y) coordinates for an element cannot be determined for a selected element within a grid square. CARMONETTE is capable of approaching the degree of resolution desired by representing the unit as an individual soldier or vehicle. However, because CARMONETTE is restricted to forty-eight

units on each side, the resolution attainable is again below company level. In conclusion, there is a need for a dismounted infantry model capable of capturing the actions and effects that the individual soldier has on the battle at the battalion and company level. The models described can assist in analyzing the detailed technical questions up to platoon level yet require varying degrees of simplification or aggregation in order to model combat at the battalion level. In either case, the synergistic effects of each soldier on the outcome of battalion combat cannot be measured. With this requirement in mind, research is being conducted in the enhancement of the STAR model to include the play of dismounted infantry soldiers.

III. OVERVIEW OF STAR

A. THE STAR MODEL

The STAR model is a high resolution, event structured, stochastic, two-sided simulation of the combined arms air/land conflict. The model may be run using either digitized terrain or a continuous, functional terrain representation developed at the Naval Postgraduate School. All units are resolved to the individual vehicle or soldier level. Conflicts are modelled between a Blue Brigade/Red Division echelonment, or any subset of these organizations. The model is capable of playing a wide range of resource allocation, fire and maneuver tactics easily modified by the user. Development, expansion and implementation is an ongoing effort. This section discusses the planned operational status as of September, 1980, except as specifically noted.

In the functional terrain model, the macro-terrain is represented as a series of hills which are ellipsoids in horizontal cross section and have the characteristic normal bell shape in vertical cross section. Any desired representation of the battlefield contour map may be achieved by under or over laying additional functional hill masses. Elements exist at any arbitrary coordinate on the continuous battlefield. Features such as forest, vegetation, soil types, rivers, built-up areas, obstacles, minefields,

etc. are represented by geometric overlays on the macro-terrain. These features may be created or destroyed as required. Line of sight and mobility are precisely determined analytically, because of the functional nature of the representation.

The ground model is capable of playing the full range of armor/anti-armor systems, to include dismounted ATGM's (operational in December, 1979). The basic event sequence of acquisition (based on time to detect), target selection, fire and impact characterizes the basic model flow.

The Air/ADA modules provide the model the capability of representing the two-sided play of helicopters and air defense systems, as well as close air support fixed wing aircraft. (The A-10 and high performance aircraft modules are scheduled for completion in March, 1981). A full range of firing tactics and acquisition modules are played, to include indirect missile firings and hand-off illumination when required. All air and ADA systems are explicitly portrayed on the high resolution battlefield, to include the FARRP for the aircraft.

The Field Artillery portion consists of six basic modules, all of which are explicitly represented on the battlefield. The FO module generates requests for fire, categorized as FPF, trigger areas, and clusters (for targets of opportunity). The allocation module and fire direction module are capable of playing a range of target

prioritization capabilities, including TACFIRE. The gun module and assessment module provide the capability to represent either volley or individual tube fire and assessment as required by munition and supportive data. The counterbattery module represents appropriate radar acquisition and target selection methodology. The Artillery, Air and ADA models are closely linked to assure coordination of activities.

The Limited Visibility/Smoke module provides a dynamic representation of the physical environment which continually changes as a function of appropriate parameters. Smoke clouds are dynamically created, moved and dissipated during the battle. Sensor physics are employed to describe a wide range of devices and to determine environmental attenuation. Finally, times to detect are determined from physical considerations, as well as the mode of sensor employment.

The Communications/Electronic Warfare module provides explicit representation of all artillery nets, as well as the ground unit tactical nets. A spectrum of two sided EW methods are also represented. Ongoing development of modules for radar countermeasures, ADA Command and Control, air request nets, red DF/artillery nets, and the remote designators will result in a total COM/EW capability for the model.

The Supression module represents the effects of direct and indirect fire on the delay of element functions such as

detection, firing, and movement. Current lack of data requires that parametric estimates of supression factors be provided as module inputs.

Initial design of a high resolution ammo/fuel logistics module (individual carrier level of resolution) which dynamically represents the logistics/combat interactions from brigade trains to the actual resupply of the combat vehicles is complete. Implementation of these modules is an ongoing development effort.

B. THE USE OF SIMSCRIPT IN STAR

The Simulation of Tactical Alternative Responces (STAR) Model achieves a high degree of resolution, yet maintains tranparency to the model user. This is accomplished primarily through the use of the SIMSCRIPT II.5 language.

The SIMSCRIPT language is designed to model discrete event simulations. The language is very readable in the sense that the command structure is much closer to English than that of many other languages. SIMSCRIPT is further enhanced by a compiler which provides error messages and traceback routines which assist greatly in the debugging phases of the modelling process. The basic elements of SIMSCRIPT are entities, attributes, sets and events.

Entities are, by definition, program elements that exist in a modelled system. In STAR the soldiers, BMPs, IFVs, Tanks and other weapon systems are examples of entities. Each entity is differentiated from other entities

by the values assigned to its "attributes". All entities in the same entity class have the same attribute names but the values of these attributes differ, corresponding to how they are set by the program. Attributes can have real, integer, or alphanumeric values. Appendix F defines the entity attributes used in STAR.

A set is a group of entities with some common property. STAR primarily uses sets to denote membership to tactical organizations such as a Brigade or Battalion. This proves useful in the modelling of tactics which are unit specific. An example of this is a company withdrawal for which all members of the company set move to a new position. An entity may belong to as many sets as the programmer desires.

An event is an occurrence which takes place at a specified time, and results in changing the values of entity attributes, removing or adding entities to sets, and/or the scheduling of further future time. events. Events may be thought of as subroutines to be executed at some future time. Events take place instantaneously and do not consume simulated time.

Each entity in STAR is modelled to reflect a flow of activities over time. In particular, each entity initiates or undergoes search, detection, target selection, firing and impact. These five events are scheduled dynamically based on the current tactical situation or by way of an appropriate probability distribution. When an event is scheduled, the

SIMSCRIPT timer makes a record of the time that the event is to occur (in terms of overall simulated time) and the arguments of the event. Other characteristics of the event may be recorded in a manner similar to the assignment of attributes. At the appropriate simulated time, the event is executed unless cancelled by some logic provided by the programmer.

C. SELECTED STAR MODULE DESCRIPTIONS

The STAR model uses a functional approach to modelling continuous macro-terrain. This technique results in a terrain board which is composed of a series of elliptically shaped hills. This representation provides the following benefits over a digitized representation:

1. The terrain is represented at every point in space.
2. Functional terrain provides a great core-storage saving compared to the requirements of a digitized representation.
3. It is possible to characterize terrain according to such criteria as the number of hills and steepness in order to generate "typical" terrain for an area for use in future analysis.

Movement in STAR is a function of the element's speed, direction of movement, the acceleration or deceleration permitted by the terrain slope, and the selected movement tactics. The movement routes are modelled as a series of linear segments. This technique is executed by the input of

(x,y) coordinates which map the route. The individual elements move along their designated routes from battle position to battle position. STAR does not depend on the interpolation or extrapolation of an individual's position in order to satisfy a required computation.

The target acquisition process in STAR begins with the representation of the entities on the STAR battlefield. Each element is initially assigned a search direction. In the absence of a stimulus in the primary direction of search, the observer enlarges his search envelope in an effort to acquire a target. If line of sight exists between opposing elements, a time to detection is calculated. The detection is scheduled to occur in the assessed time to detect. The element is then placed on a list of detected elements which is maintained for each entity in the battle. Elements can also be detected as a result of a firing stimulus detection. This models the phenomena of acquiring a target as a result of seeing the signature of the firing of a round. If the observer is looking in the right sector at the time of fire, he obtains perfect sector knowledge pertaining to the location of the target for one event. This results in a decreased time to detection.

After a target has been selected by the firer to engage, the fire event takes place. The direct fire module utilizes bias, dispersion and lethality data from AMSAA, BRL and appropriate field tests. STAR updates the target's

location and disposition at the time that the shot is fired as well as at the time that the impact of the round occurs. The impact occurs at the appropriate time after the fire event. The outcome of the impact is assessed in terms of the full range of kill types.

After the round has impacted, the firer makes a decision whether to fire again or to go to full defilade. This decision is a function of the WE.HIT/WE.MISS tactic chosen by the user. The amount of time the firer remains in defilade is also a user input.

The module descriptions stated in this chapter are designed to be brief. A more in-depth discussion of the characteristics of the STAR model may be found at Appendix A in an untitled paper presented by Lieutenant Colonel Edward P. Kelleher Jr. to the 43rd Military Operations Research Symposium in West Point, New York in June 1979.

IV. GENERAL METHODOLOGY OF THE DISMOUNTED STAR MODEL

The dismounted infantry version of STAR is structured to support the basic model discussed in Chapter III. The model is conceptually capable of simulating a Blue Brigade versus a Red Division, or any subset of these organizations. Current versions of the model are used to simulate a company versus a battalion. In any case, all units are resolved down to the individual soldier level. To utilize the model the user is required to design the composition of opposing forces down to the individual soldier level. Additionally, the user is required to design a scheme of maneuver for the offense and the defense. This includes a list of battle positions, routes, formations and decision logic used to control movement of units. Within battle positions the user selects locations of vehicles and each infantry soldier.

The infantry soldier is modelled as an individual entity. Figure 4-1 is an example of how the model portrays the physical characteristics of each soldier. The body is subdivided into six sections for the purposes of casualty assessment. These sections are the head, the thorax, the two arms, the abdomen and the pelvis and lower extremities.

Different types of infantry soldiers are distinguished according to the primary weapon they are assigned. The following primary weapon types are currently simulated in the model:

ENTITY REPRESENTATION

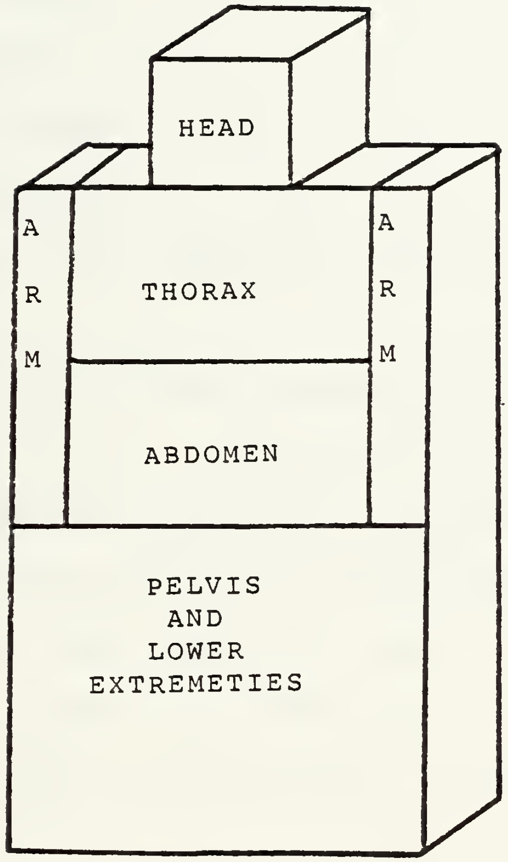


Figure 4-1

a. M16 Rifleman

b. M60 Machinegunner

c. SAW Machinegunner

d. M203 Grenadier

e. Dragon Gunner

f. RPG Gunner

g. AKMS Rifleman

h. PKM Machinegunner

Basic loads of ammunition are assigned to each infantryman.

Each soldier is allowed to carry six different types of ammunition. The types of ammunition a soldier carries is related to all weapons a soldier is assigned. As an example, all M203 Grenadiers assigned 5.56mm, M203HE, M203DP, LAWs and handgrenades. The amount of each type of ammunition assigned to a soldier is determined by the user. Each soldier's basic load is decremented as he expends rounds in the simulation.

The tactical structure modelled is the real tactical structure of a unit. Every soldier is assigned to a squad, a platoon, and a company and is linked to a battalion and a brigade through his company and battalion commanders respectively. The composition of each squad is variable and determined by the user. When modelling mechanized infantry each squad member is assigned to an infantry fighting vehicle.

When the dismounted model is used to simulate a unit in the defense, such as a company, opposed by a unit in the attack, such as a battalion, the user is required to formulate a plan for defense in the same manner as preparing an operations order. An initial position is selected to defend. Subsequent positions for the defense must also be selected if the defending commander determines that it may be necessary to move. The routes used by each platoon to move to a new battle position must be input by the user. Figure 4-2 is an example of two battle positions (AREA 101, AREA 201) and a preselected route (ROUTE 1). All routes are designed as a series of one or more linear segments. Route 1 in Figure 4-2 is comprised of three segments. Each segment connects two Movement Control Points (MCP). A route may contain as many MCPs as necessary to obtain the desired movement pattern. In addition to specifying the route a platoon travels on, the user is also allowed to select the formation the platoon is to use on each segment of the route. This allows the user to select a formation to correspond to the terrain. The number of formations and the types of formations in the model are also determined by the user.

At each battle position the user is required to input to the model the coordinates that each vehicle and infantryman in the platoon is assigned. While in a defensive position each soldier is assigned a primary sector of fire.

MOVEMENT IN DISMOUNTED-STAR

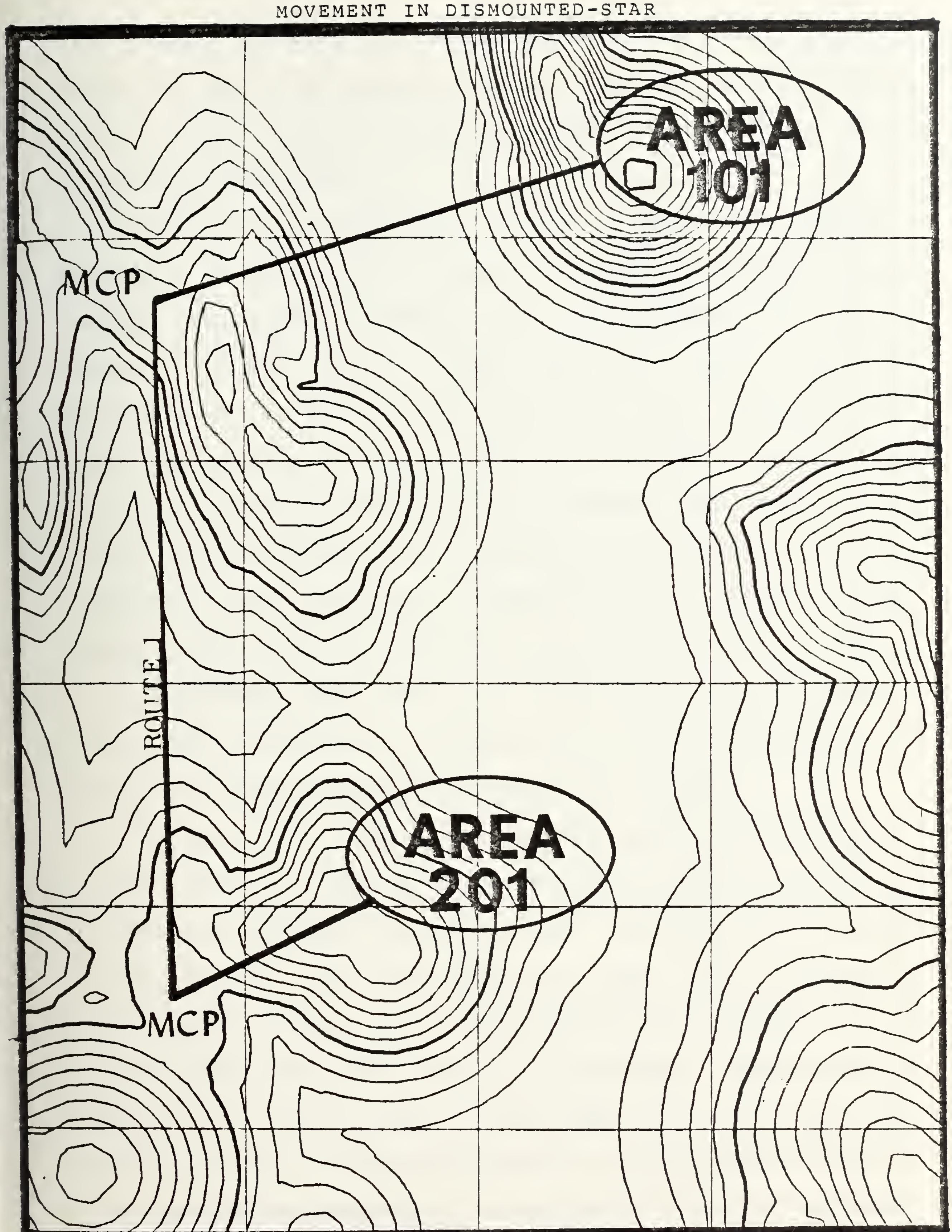


Figure 4-2

Infantrymen attempt to detect targets in their primary sector of fire. If targets are not detected in the primary sector, the soldier's search sector is increased until targets are detected.

Each soldier maintains a list of detected elements. From this list the firer selects one of the elements as a target. The element selected is that element that ranks highest on the firer's target priority list. The user designates a target priority list for each infantryman by primary weapon type. As an example, each M203 Grenadier has the same target priority list. Targets are prioritized according to the range and the type of target. Additionally, the weapon type and type of ammunition to be fired at each target is specified by the user. As stated before, given a list of detected elements, the soldier selects that element which ranks highest on his target priority list for which he has the appropriate ammunition available to fire.

The number of times a soldier fires before taking cover is designated by the user as a tactic. The tactic specified for each weapon type also determines the action to be taken if the previous shot was a hit or a miss. If the firer is not allowed to remain in a firing mode he is placed into a full defilade position. There are currently three tactics that define if and when a firer goes to a full defilade position. Tactic 1 allows the soldier to continue to fire as a function of the number of catastrophic kills or the total

number of shots. Tactic 2 requires the firer to continue firing without ever going to full defilade. Tactic 3 allows the soldier to fire as long as he has not exceeded a specified number of shots since the last time he was in full defilade. The user also controls the amount of time a firer remains in full defilade once he attains that condition.

The dismounted model utilizes two methods of casualty assessment. The first method uses data such as the probability of hit, probability of kill, and the other kill types. The second method of casualty assessment uses accuracy data to determine whether a round strikes the target. The degree of incapacitation is then determined separately. The type of projectile fired determines which of the two methods is used to assess casualties.

If the projectile is a LAW, VIPER, M203DP or RPG the data includes the probability of mobility kill, firepower kill, and catastrophic kill for target vehicles. Similar data is used to calculate expected casualties to infantrymen mounted in the vehicles. These assessments are determined by Monte Carlo procedures.

For small arms non-fragmenting rounds fired against exposed targets the second procedure is used. First a target and weapon type and appropriate indices for accessing the lethality data are determined. The factors affecting the outcome are:

a. Range to the target

b. Defilade status of the target

c. The number of rounds fired

d. The aspect angle of the target

These factors provide the indices required to look up the parameters to determine miss distances. As is usual, errors in elevation and deflection are assumed to be independent, and normally distributed. Using standard Monte Carlo techniques, the miss distance from the impact point is computed in elevation and deflection. For burst fire weapons, the displacement of burst center of impact is then calculated, again assuming normality. Finally the miss distances for each round after the first round are calculated. The product is a pair of miss distances for each round fired. This results in an impact pattern as depicted in Figure 4-3.

The result of each round is then assessed individually, considering the size and aspect angle of the target, how much of the target is exposed, and the size of the miss distances for the round. The result of hitting the target may be played either simply or in great detail based on the user's desires.

If simple detail is desired, all hits are recorded as catastrophic kills. When more detailed casualty assessment is desired, the number of rounds impacting on the body and the area of impact are determined. This information is used

DISPERSIONS ASSOCIATED WITH BURST FIRE WEAPONS

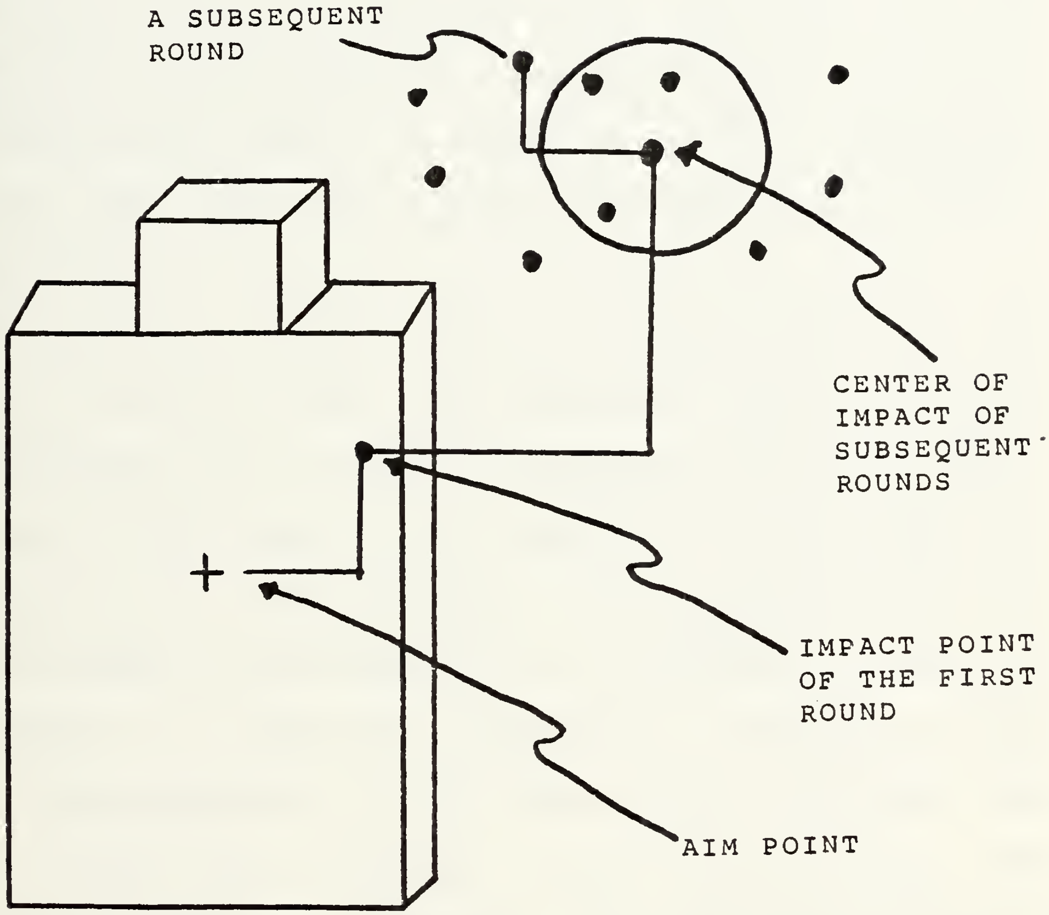


Figure 4-3

to determine if and when a target is incapacitated. There are varying types of incapacitation. These types are defined as mobility, firepower, mobility and firepower, and catastrophic incapacitation. The time until incapacitation is determined based on a total body probability of incapacitation by a specified time (a tier of the incapacitation input data). If incapacitation is to occur the incapacitation event is scheduled at the appropriate time. Until that time, the soldier continues to function in a normal manner.

The casualty outcomes have a direct effect on the movement of units in STAR. The movement decision criteria are determined by user input thresholds for attrition level, force ratio, and range to the enemy. If at any time the threshold for a weapon system is exceeded, the unit commander checks to determine if he is allowed to move to a new position. If the unit commander is allowed to move without the permission of the next higher command, the movement takes place. However, if the unit requires the permission of the higher commander, the decision to move is based on the decision criteria of the higher commander. In addition to the attrition levels, a minimum range can be specified which, if reached by the enemy, causes the defending unit to move.

If a unit determines it is time to move, all dismounted

entities search for a vehicle to remount. The first vehicle to which a soldier attempts to move is his assigned squad vehicle. If the assigned squad vehicle is capable of moving in the battle, squad members begin to move towards that vehicle to remount. If the vehicle suffers a mobility kill or a catastrophic kill while the squad is moving to the vehicle, squad members stop and search for a new vehicle. The platoon is searched to determine which vehicle is closest to each soldier. Soldiers then begin moving toward that vehicle. Vehicles wait a specified amount of time for the infantry to arrive to remount. The time a vehicle waits is determined by the user. Any soldier that does not arrive at a vehicle before it departs, attempts to move to the next defensive position on foot, along the route preselected for the platoon. When a vehicle arrives at a new defensive position, it moves to its assigned vehicle position. At this time the soldiers dismount from the vehicle. Each soldier then moves on a straight line to its assigned infantry position.

A general plan of attack must also be planned for the attacking forces. Attacking units use preselected routes and formations. The actions an attacking unit can take during the attack are determined by the user through the use of attrition levels and force ratio thresholds. Actions available to the attacker are continue to attack mounted,

dismount, go into a hasty defense, or withdraw. As an example, the user may specify that an attacking unit is to dismount all forces if during the attack the force ratio is below 2.5 or the attacker loses two of his T72s.

When a determination is made to dismount and continue the attack, all vehicles carrying infantrymen stop to let them dismount. Once a dismount takes place, soldiers move to the front of the vehicle. The squad then assumes a line formation and assaults the objective. An example of this action is found in Figure 4-4. It is important to note here that additional work is underway to control the dismounting forces as a function of the following:

a. Force ratio

b. Attrition rate

c. Distance to the objective

d. Predesignated dismount check points

A final control measure used to control movement of soldiers and vehicles is the use of speed limits, and rates of acceleration and deceleration. Periodically during any move, the slope of the terrain on which the vehicle or soldier is travelling is calculated. The soldier's movement rate is then adjusted to correspond to the terrain. The terrain is classified as either "upslope", "downslope", or "level". Each vehicle type is assigned two different sets of speed limits. One set is used when the vehicle is moving

DISMOUNT OF ATTACKING FORCES

NOTE: Elements dismounting do so to the immediate front of their vehicle. These elements always dismount perpendicular to the direction of movement of their fighting vehicle. (Picture taken from FM 23-3, Tactics, Techniques, and Concepts of Antiarmor Warfare, August 1972, page 55)

Figure 4-4

in a mounted attack. A second speed limit is furnished which reduces the speed of the vehicle when moving with a dismounted force.

This concludes the general discussion of the methodology used to model dismounted infantry in the STAR model. A more detailed discussion of the methodology to include the mathematical models and examples of input requirements is provided at Appendix B. Routines and events modified in the ground model are reviewed at Appendix C. The new routines and events developed for DISMOUNTED-STAR which implement the methods discussed in this chapter are at Appendix D.

V. FUTURE MODEL ENHANCEMENTS

The dismounted version of STAR presented in this thesis presents a solid foundation on which to expand the high resolution simulation of both mounted and dismounted infantry combat. As an extension to the existing STAR combat model, it represents a major step in the development of a viable combined arms model. If a higher degree of resolution is needed, there are several enhancements that must be made to the model. This chapter delineates several of these which are of near term interest. Further, the areas for possible long term research are noted.

The primary short term focus is to enhance the infantry model to include the process functions being added as described in Chapter III. The first of these is the implementation and testing of the fragmentation assessment routines. The play of artillery, mortars and grenades is essential to achieving the degree of resolution needed to assess the dismounted battlefield. The routines to assess these phenomena have been written, but have not been implemented and tested in the model as of June 1980.

Other areas needing further work in the near term are the following:

1. Infantry fighting positions as currently represented do not have overhead cover. This becomes important with the implementation of the fragmentation

assessment routines described above.

2. Attacking soldiers do not assault by using short rushes. They do take advantage of the available cover, but the attack and movement are continuous. The short rush, seek cover, short rush tactic should be modelled.

3. Some work has been completed to expand the criteria for dismounting elements enroute to an objective to include reasons other than attrition or force ratio. These enhancements will allow the user to specify preselected dismount points which can be either mandatory or optional dismounting areas, depending on the tactical situation. This is intended to portray the commander's pre-mission terrain assessment process in determining suitable dismount points for selected objectives. Additionally, the user will be able to designate the size, composition and location for the forces to overwatch the assault by dismounted forces. This work is of immediate interest and should be completed.

4. The phenomena associated with the close-in battle should be simulated in a more realistic manner. The improvements here are in terms of:

   a. Obstacle and minefield play

   b. Use of claymores and hand grenades

   c. The Final Protective Fire structure and orchestration

5. The model should be improved in the area of Command, Control, Communications and Intelligence play to allow the elements to receive information from the scouts, higher headquarters, radars and other combat support elements.

6. An interface with STAR-AIR appears to be feasible and may enable the simulation of airmobile forces on the battlefield, as well as the effects of close air support on the dismounted battle.

Potential long term research includes the use of results obtained from the high resolution company/battalion models to develop an aggregated model which portrays combat at higher levels. A second area of possible long term interest is the modelling of urban warfare. The structure of the STAR model and the associated use of the SIMSCRIPT language make this a viable area for future research. A third possibility is the analysis of scenario specific combat situations. The Rapid Deployment Force (RDF), amphibious warfare, and two-sided second echelon infiltration scenarios can be analyzed in the long term. Finally, there is a need for a combat model capable of assessing the effects of chemical and nuclear warfare on dismounted infantry tactics. Again DISMOUNTED-STAR has the potential to serve as an appropriate tool to assist in this analysis.

APPENDIX A. INTRODUCTION TO STAR

The following is an untitled paper presented by Lieutenant Colonel Edward P. Kelleher Jr. to the 43rd Military Operations Research Symposium (MORS) at West Point, New York in June 1979.

The purpose of this paper is to acquaint the reader with the STAR combat model. STAR stands for Simulation of Tactical Alternative Responses. The model is being developed at the Naval Postgraduate School, primarily by students with some assistance from faculty members. The origin of the model was in 1976 when a student at the school devised what he thought was a novel and interesting way to model terrain. Major Chris Needels modelled the terrain, which he called parametric terrain, as a series of bivariate normal hills. As his thesis, he built that terrain model and devised a line of sight algorithm for the terrain. The choice of the word "parametric" was unfortunate, since that word had been previously been used to describe an entirely different type of terrain model and conjures up an image which is inaccurate. A better way to describe the model is that it is a functional representation of the terrain, as will be discussed later in this paper.

The next step was a Blue Field Artillery versus Red maneuver elements model that was essentially a duel, built

on the terrain model in 1977. It played a single Blue Field Artillery battalion firing in support of a Blue force facing a Red regimental attack. No Counter-Battery fire was played.

In June 1978, a serious attempt to build a combined arms model was initiated. In December 1978, the initial battalion level runs were made. In those runs, made in support of the XM-1 stowed load study, 321 individual elements were played. Those elements were tanks, BMP's, Infantry Fighting Vehicles, Dragon teams, Improved TOW Vehicles and M113's. The battle played was a direct fire battle only. Although there was a Field Artillery model essentially working at that time, initial runs were made without that module. The battle took place on a ten by ten kilometer functional terrain box.

In February of 1979 the effort to expand the model began by replacing the ten by ten kilometer terrain box with a ten by thirty terrain box and modifying the logic to permit the play of a defending Blue Brigade facing a Red Division-sized attack.

As of September 1979, the status of STAR is that the battalion model is up and working on the ten by thirty kilometer battlefield, which represents the terrain between Hunfeldt and Bad Hersfeldt. A parametric suppression model is included, as is a simplified resupply model. A Communications-Electronic Wareface module was built as a

thesis in late 1978, and has been run with the battalion model. It has not yet been incorporated into the brigade model, but plans exist to reinstall the module. An Air-Air Defense module has been debugged and installed, and is in process of integration debug runs. The Field Artillery module is in process of substantial revision to make it more realistic, and to attempt to capture the synergistic effects of the indirect fire contribution to the combined arms battle. Limited visibility and smoke modules are being developed by the U.S. Army Night Vision Laboratories to be installed in the STAR model. Brigade test runs were initiated on 1 June 1979. The planned course of development is to develop the brigade modules on the ten by thirty kilometer box, and then to simulate the Division 86 scenario on a thirty by sixty kilometer box in the same general area. The intent is not to evaluate Division 86, but to implement an approved scenario, since that is substantially easier than developing a totally new scenario.

One of the modules which sets STAR apart from other combat models is the functional representation of terrain. Any medium to high resolution combat model must provide a terrain model in which the representation of terrain permits Line of Sight (LOS) computations between any two points on the battlefield, realistic limits on vehicle movement, and the capability to model terrain related features, such as forests, rivers, built-up areas, and so on. It is also

desireable that elements be permitted to exist at any point on the battlefield. The goals of the STAR terrain were to attain the ability to represent a large enough battlefield to realistically model the brigade combined arms battle, with all elements actually on the battlefield; to achieve a representation of sufficient resolution to permit detailed calculations of movement and LOS for individual elements; and to permit the flexibility to model a variety of terrain features and their influence on the battle. An assessment of the size battlefield needed indicated that digitized terrain would not meet these goals because of the significant on-line core storage required. To store the elevations on a thirty by sixty kilometer battlefield at 100 meter intervals requires 180,000 words, or on an IBM machine, 720,000 bytes. This requirement is prohibitive on most Army computers.

The functional terrain model used in the STAR model represents the terrain as a series of hills, which are ellipsoids in horizontal cross section, and have the characteristic normal bell shape in vertical cross section. The hills are no longer directly bivariate normal, due to the difficulties with the shape when the correlation coefficient approaches 1.0, as it must for hills or ridge lines which run northeast to southwest, or northwest to southeast. Each hill is represented by seven parameters, which determine the height of that particular hill at any

point on the battlefield. To determine the height of the terrain at that point, the height of each hill that has any influence at that point is calculated. The maximum of those heights is the terrain height. Figure 1 shows an example of this terrain, with three hills. On the left side of the figure, Hill 1 determines the height of the terrain, in the center, Hill 2 is the dominating feature, and determines the terrain height, and on the right Hill 3 determines the height. The result is a terrain surface which is directly represented at every point on the battlefield. The method by which the terrain model is generated is still something of an art. It consists of beginning with dominant terrain features, fitting ellipses to them, moving and rotating them until a "good" representation is achieved, then moving on to smaller features and melding them into the overall terrain. When assessing the validity of any model the question that must be addressed is: "What is close enough? What is it that we need from this model? Can we get the level of detail that is needed?" In the opinion of the STAR modelling team, for the STAR terrain model, the answers to those questions are "Yes".

Figure 2 is a reproduction of the terrain contours of a ten by ten kilometer piece of terrain in the developmental ten by thirty box. A qualitative feel for the fidelity of the model may be achieved by comparing Figure 3 to Figure 2. Figure 3 is a computer generated map of the contours of the

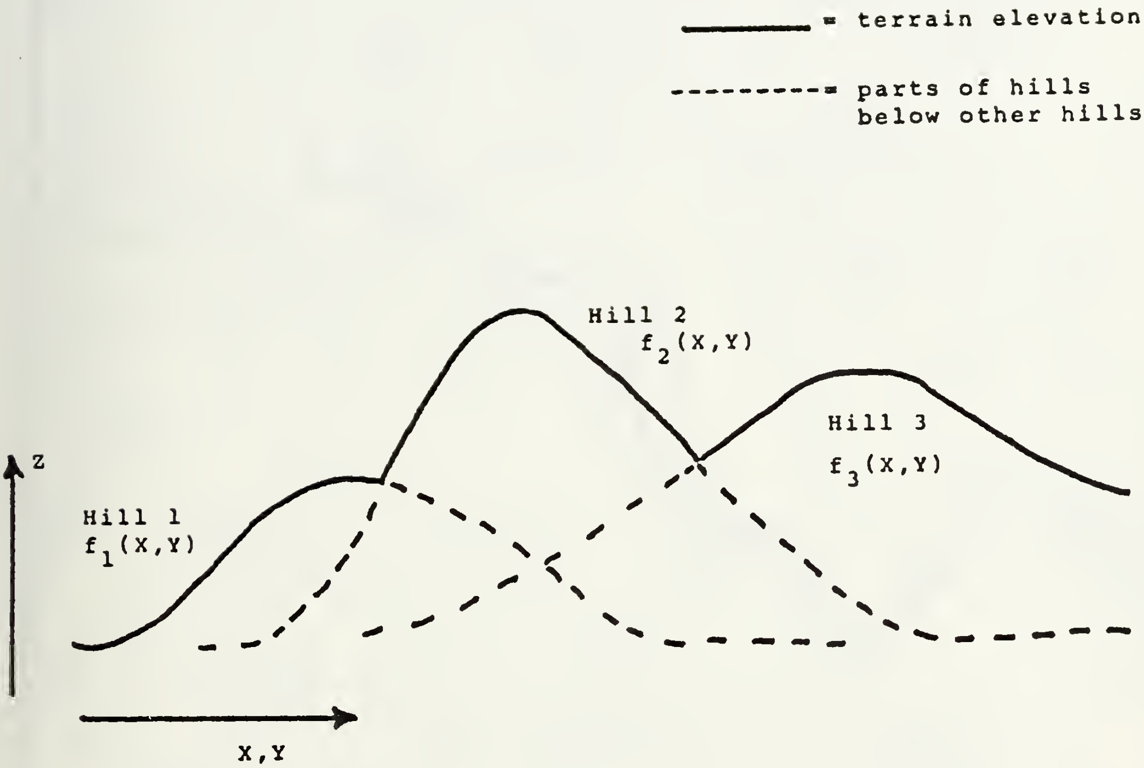


Figure A-1 Functional Terrain

TOPOGRAPHIC CONTOUR MAP

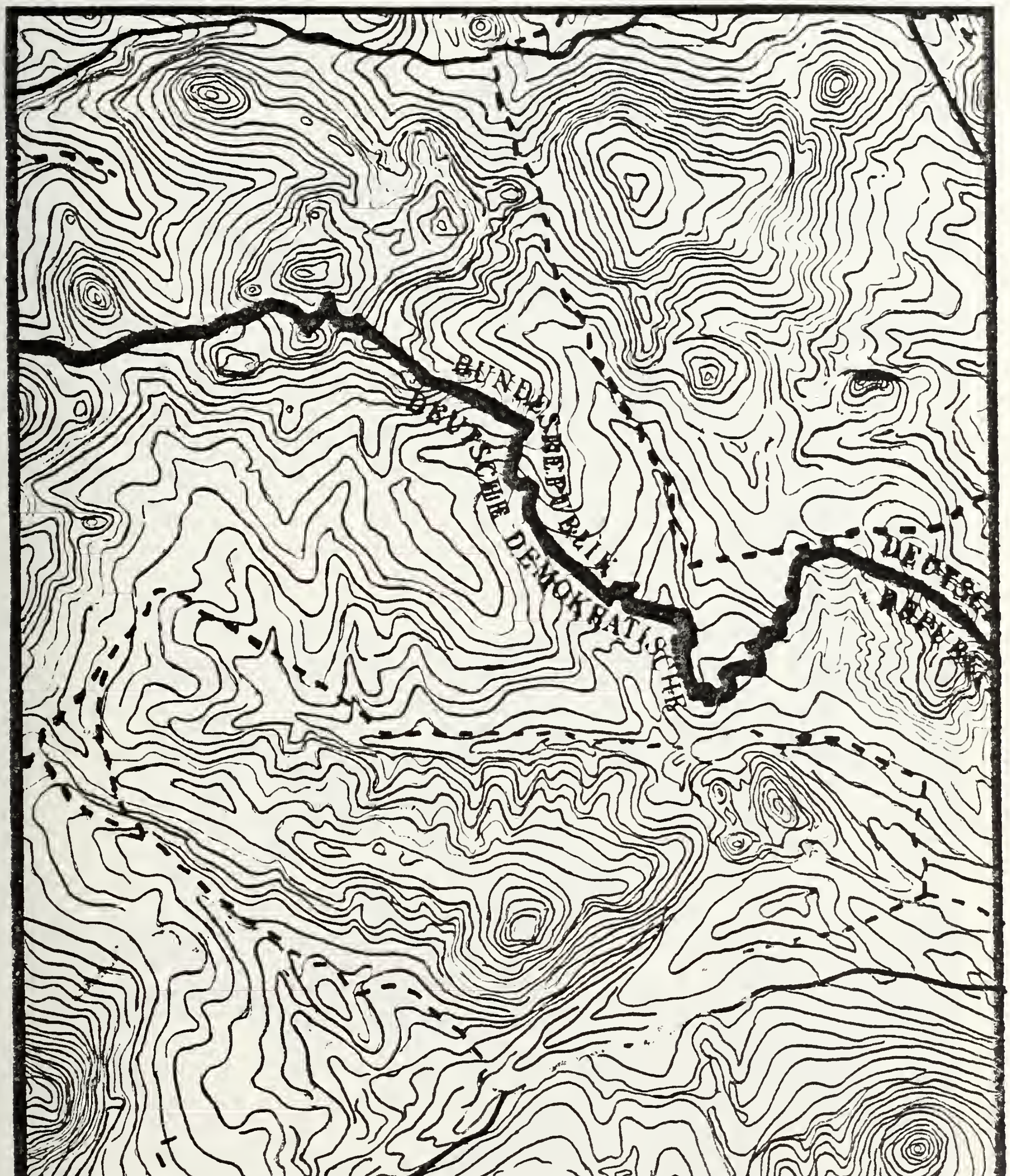


Figure A-2

FUNCTIONAL TERRAIN CONTOUR MAP

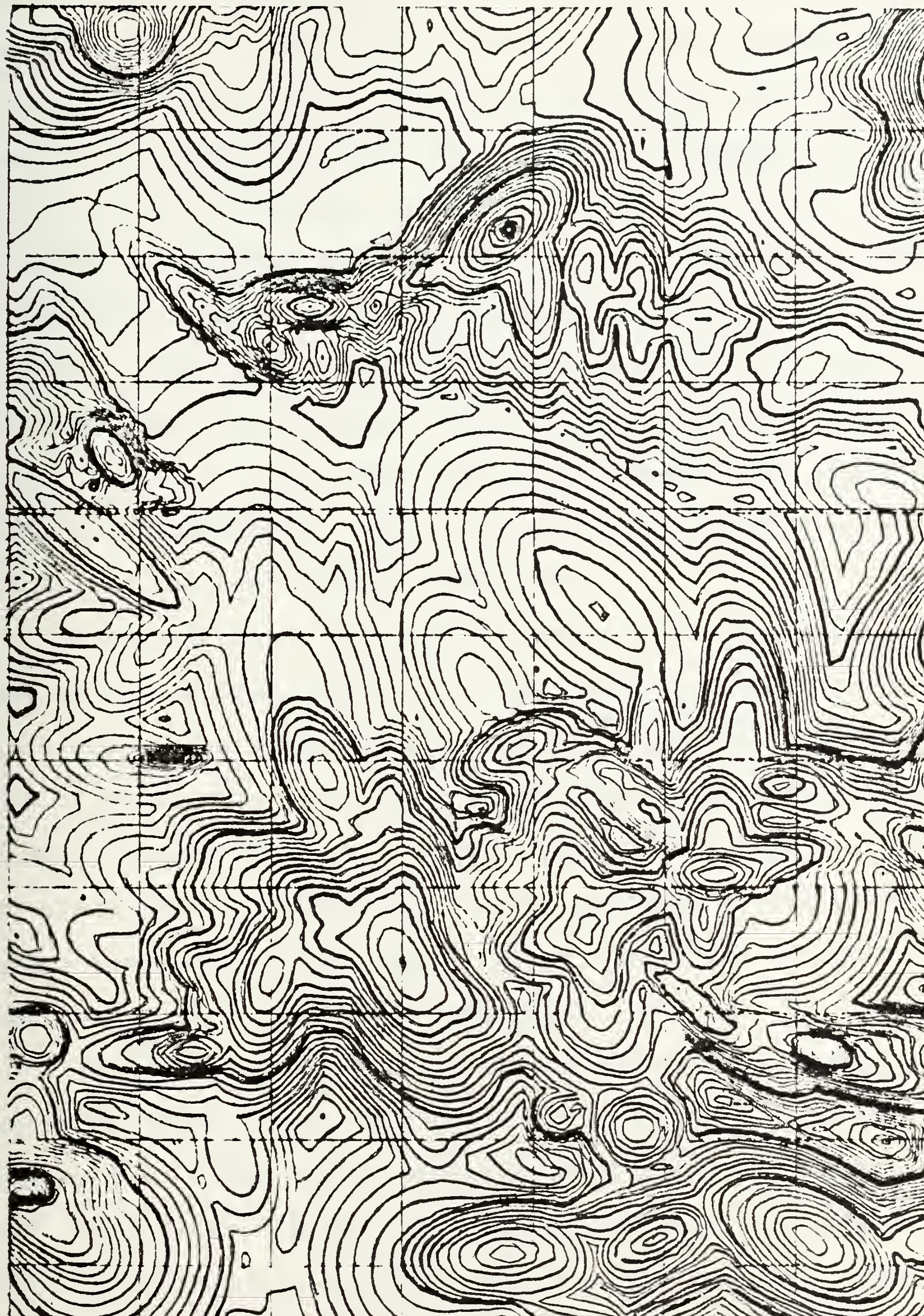
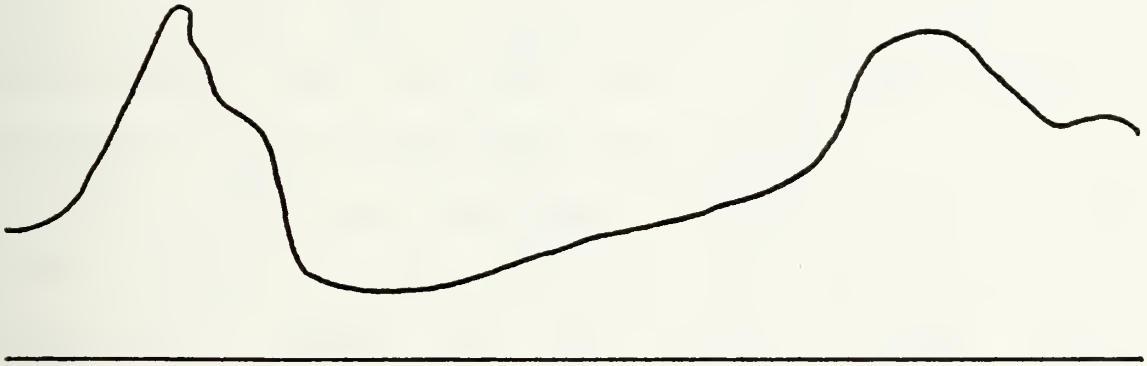


Figure A-3

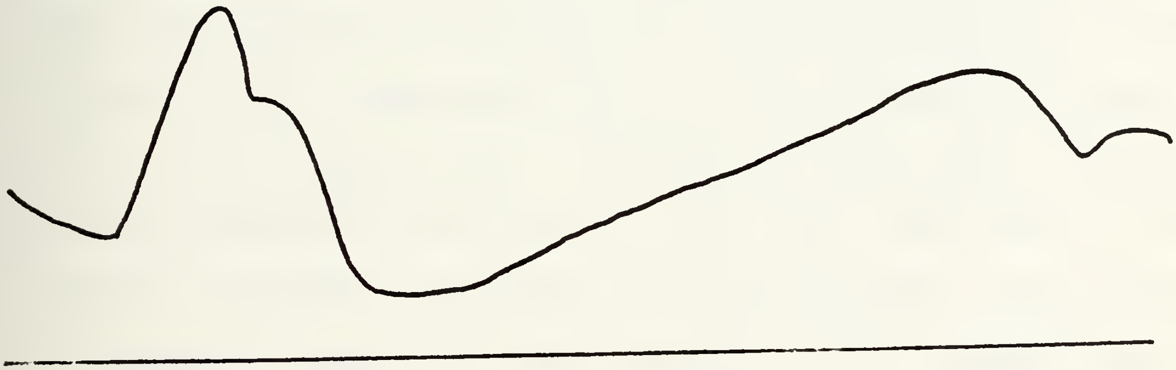
STAR terrain model of the same area. The reader's attention is directed to the hill structure in the East-Northeast portion of the box, to the north to south ridgeline just east of the center of the box, to the three generally round hills in the northwest corner, and to the more gently sloping terrain just west of the central ridgeline. In our opinion the model provides a "good" representation of the terrain. While extensive comparisons have not been made, it is known that the functonal terrain model does not yield the same line of sight fans that the equivalent digitized terrain model does. It is not known, however, which is closer to the real Line of Sight fans.

A second feel for the closeness of the representation may be gained from Figure 4, which shows two vertical cross sections of the same terrain. Figure 4 was constructed by a U.S. Army surveyor from the contour map along a preselected five kilometer straight line. Figure 4b is a computer generated vertical contour of the same five kilometer line, using the STAR terrain model. A 10:1 vertical exaggeration was used to facilitate the comparison.

The most frequently voiced question about the STAR terrain model, involves the level of effort involved in coding the hills. Our experience is limited to a sample of two individuals. Both are trained U.S. Army surveyors, in the grade of E-4, one with a college degree, one without. It took each of these men approximately four weeks to code



4a. Contour Map



4b. Functional Terrain

Figure A-4 Vertical Cross Section

their first ten by ten terrain box. At the end of six weeks, each was at a point where he could code the hills for a single ten by ten box in five to seven working days. That indicates that for a thirty by sixty terrain box, approximately four and one half man months would be required, after the individual was trained.

One of the most important functions that a terrain model provides is Line of Sight (LOS) calculations. The existence or absence of line of sight between opposing forces profoundly affects the course of the battle. Figure 5 is a representation of an observer attempting to see a target. The STAR Line of Sight (LOS) Model computes LOS beginning at the top of the observer. In Figure 5 the target is partially concealed by the micro-terrain in the vicinity. This characteristic is an attribute of every entity on the battlefield: How much of the entity is above the surrounding micro-terrain, and thus available to be observed. The STAR LOS model first determines how much of the target is above the micro-terrain. This establishes a maximum fraction of the target which may be visible. The model then checks every hill between the observer and the target to determine whether that hill decreases the fraction visible. If, at any point, the fraction goes to zero, line of sight does not exist and the computations are ended.

In Figure 5, Hill 1 is checked first. Since it is lower than both the observer and the target, it can not

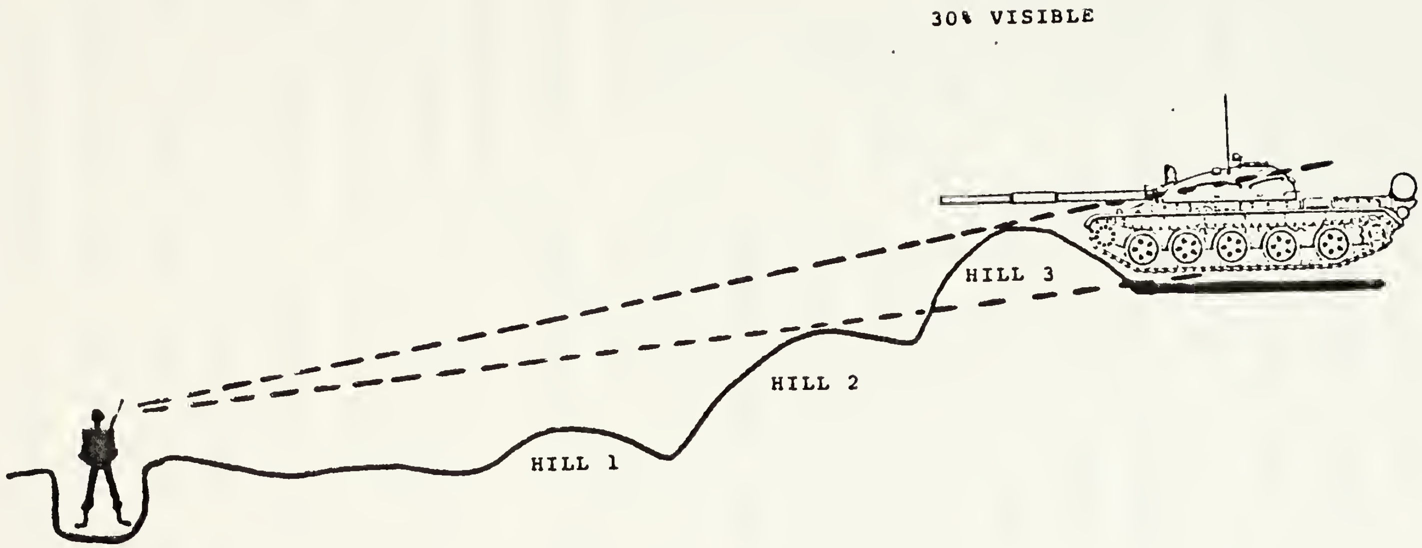


Figure A-5. LOS Example

possibly decrease the fraction visible. Next, Hill 2 is checked. Hill 2 is higher than the observer, but lower than the target, so the fraction visible may be decreased. The model computes the line of tangency between the LOS and the hillside to determine whether a decrease occurs. For Hill 2 no decrease occurs. Next Hill 3 is checked. Here the computations cause a decrease in the fraction visible. When all pertinent hills have been checked, the remaining fraction visible is the amount of the target actually available tor the observer to see. That number is returned to the tactical routines. Note that this is not a one-zero result. It is the actual fraction of the target available to be seen. It is worth noting, also that the location of the point of tangency between the hill and the lowest line of sight over the hill is the most difficult portion of the calculations, and must be approximately located by a method of successive approximations.

Several attributes of individual elements have been mentioned. In the STAR model the level of resolution is the individual combat vehicle. Examples are XM1 tanks, Improved TOW Vehicles (ITV), Soviet BMP personnel carriers, Infantry Fighting Vehicles (IFV), etc. Only one dismounted element is currently played, the two man Dragon anti-tank missile team. Each of these elements is described in terms of 103 attributes. Table I is a partial list of those attributes.

Simscript is an event oriented language, and STAR can

* 103 Attributes for Each Element *

NAME	COLOR
SYSTEM TYPE	BRIGADE
WEAPON TYPE	BATTALION
VEHICLE TYPE	COMPANY
DEFILADE	PLATOON
X	SECTION
Y	ALIVE OR DEAD
Z	MOBILITY DAMAGE
SPEED	FIREPOWER DAMAGE
DIRECTION	# TIMES SHOT AT
MICRO-TERRAIN	# TIMES HIT
PLOW STATUS	# TIMES FIRED
4 AMMO COUNTS	SEARCH DIRECTION
RELOAD STATUS	SECTOR WIDTH

PLUS 73 OTHERS

Table I. Element Attributes

1. Platoon Leader Handoff

2. Highest Priority Target

3. Highest Priority Target unengaged by platoon member, else Highest Priority Target

4. Highest Priority Target unengaged by a Company Member, else highest priority target (engage with automatic weapon fire only).

5. Highest Priority Target unengaged by a platoon member, else do not engage.

Table II. Target Selection Tactics

best be understood in terms of the flow of events that take place in representing the combined arms battle. Figure 6 represents the basic event flow in STAR. Event STEP.TIME is the event that drives the rest of the simulation. The event updates the position of each element on the battlefield, and then checks Line of Sight between each Blue combatant and every Red combatant. If LOS exists, the percent visible of the target is used to compute an "equivalent range" to a fully exposed M60 tank. That range is used, in conjunction with target crossing velocity, terrain complexity and a random number to generate a "time to detect" for that observer-target pair.

If that time is thirty seconds or less, the next event in the chain, DETECT is scheduled in the appropriate amount of time. The thirty second criteria is based on two considerations. First, in thirty seconds of a mechanized conflict many things change affecting the detection rate, and eventually the calculated time to detect will no longer be valid. Second, the detection model in STAR is basically the DYNTACS detection model. That was based on field experiment data(using M60 tanks) and is only valid for times less than or equal to thirty seconds.

If the calculated time is greater than thirty seconds, the detector tries again at the next occurence of STEP.TIME. At every STEP.TIME each element, Red or Blue, has an opportunity to schedule one or more detections.

EVENT FLOW

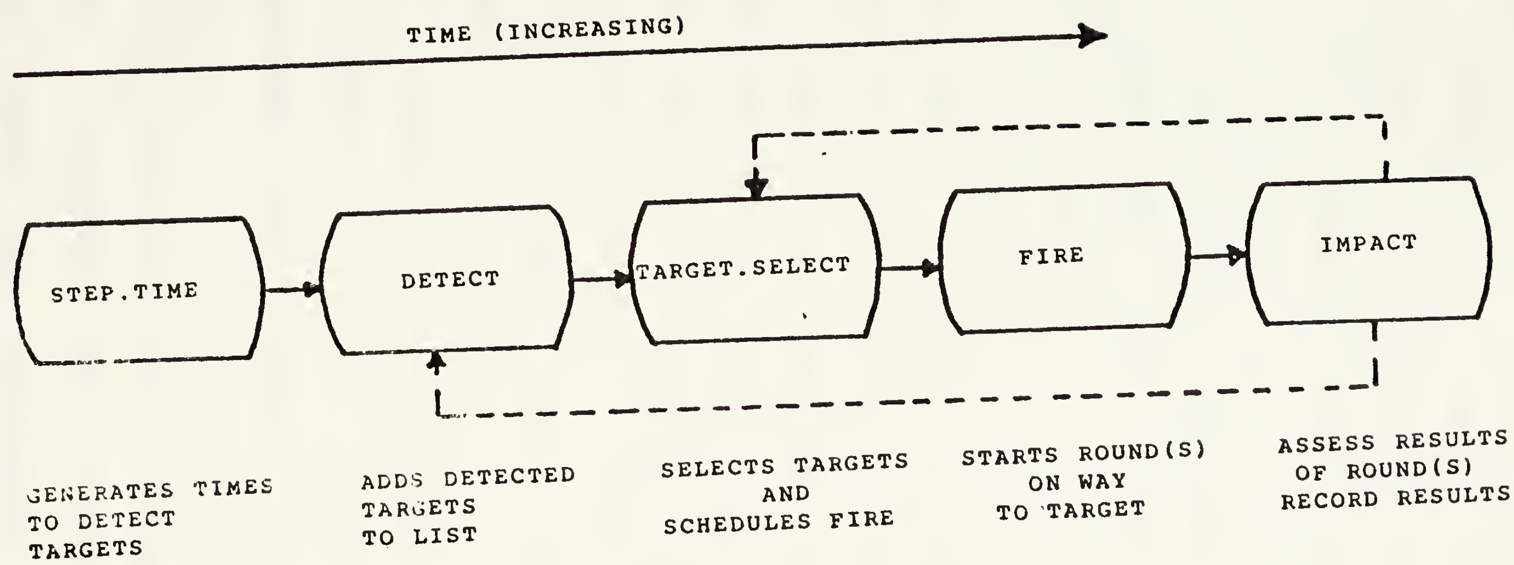


Figure A-6. Event Flow in STAR

When event STEP.TIME is completed, the simulation time is advanced until the next scheduled event, DETECT for example. Two arguments are passed into DETECT, the detector and the entity to be detected. The program checks to see if both entities are alive and if Line of Sight still exists. If these conditions and others are met the detection occurs. At this point the detected target is put on a list of targets owned by the detecting entity, and event TARGET.SELECT is scheduled.

This event does just what its name implies. If the detector is not engaged in firing, he selects the "best" target on the list as a candidate for engagement. "Best" is defined by a set of rules which are user selected and are a function of the range to the target, what type entity the target is, what type of ammunition the selector has available, and a series of rules of engagement based on the amount of positive fire control being simulated for the selector's weapon type. Table II gives an example of several rules of engagement. The first two rules in Table II are widely accepted as being used by various armed forces. Platoon Leader Handoff implies that the selector attempts to acquire the target his platoon leader is engaging. If he acquires that target he engages it. If not he does not fire. The second rule of engagement is that each entity always selects his own highest priority target. The other tactics listed are self-explanatory and reflect

various levels of fire control. Figure 7 shows an example of the combined tactic: Platoon Leader Handoff, followed by highest priority not engaged by another platoon member. If all available targets are engaged, engage highest priority with automatic weapon fire rather than main gun or missile. In Figure 7, BLUE 1 thru BLUE 4 are members of the same platoon. BLUE 1 is selecting a target. He attempts to acquire RED 2, who is being engaged by BLUE 3, the platoon leader. Lack of Line of Sight prevents this. The two targets on BLUE 1's list are both engaged by other platoon members. BLUE 1 therefore selects his highest priority target, RED 3, but engages him with automatic weapons fire only. The selection of various levels of positive fire control and fire distribution methods give the user substantial flexibility in investigating possible tactics.

When the target selection procedure is finished, if a target has been selected, the event FIRE is scheduled at a time based upon a random draw from the user specified lay and load time distributions. At the appropriate time the event FIRE is executed. Again line of sight is checked, as is the ALIVE.DEAD status of both the firer and the target. As will be discussed later, STAR differentiates between various types of damage and kill. A firer who has suffered a firepower kill or a catastrophic kill is not allowed to fire. Targets which have suffered a catastrophic kill and targets who have been mobility and firepower killed for at

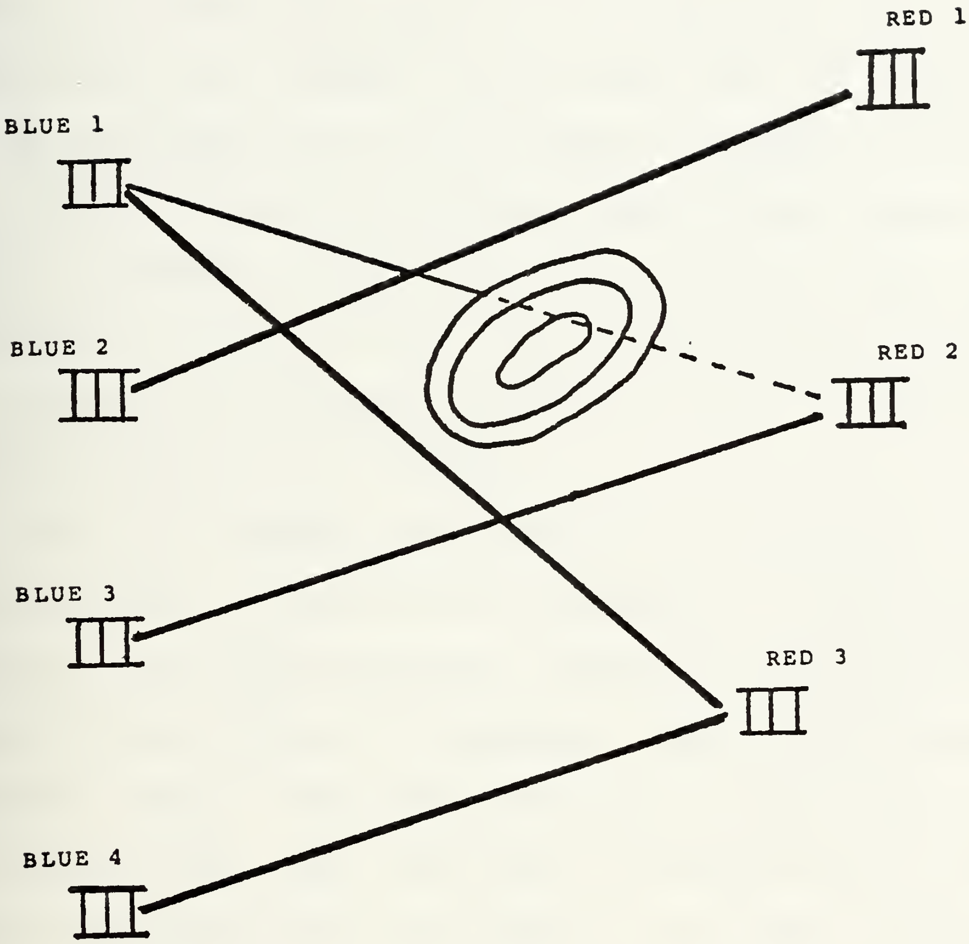


Figure A-7. Sample Target Selection

least a user specified time are not fired upon. If all the necessary conditions are met, the FIRE takes place and event IMPACT is scheduled based on the range to the target and the velocity of the fired round. The identity of the firer and target, and the percent of the target that the firer could see at the moment of fire are passed to IMPACT as arguments.

At the appropriate time the event IMPACT occurs. First LOS is checked. If no line of sight exists the round is assessed as a miss. If any part of the target is visible, the routines to determine whether or not a hit occurred are called, starting with COMPUTE.

Routine COMPUTE determines the parameters of the distributions of the miss distances in deflection and elevation for Tank main guns, antitank guided missiles, and large caliber direct fire weapons. These miss distances are assumed normal. The parameters are biases and standard deviations, and their magnitude is based upon a number of factors including type and speed of the firer, speed of and range to the target and whether or not the firer has fired at this target and sensed the preceding round. Having determined the biases and standard deviations, COMPUTE calls GEOM.

Routine GEOM draws from a normal distribution and converts the miss distance distribution parameters into a miss distance in elevation and a miss distance in deflection. Using the percent of the target visible (as

opposed to uncovered) at fire, the firer's aim point is then computed. If the "turret ring" of the target was visible, the firer is assumed to have selected the turret ring as his aim point. If the turret ring was concealed at FIRE, the firer is assumed to have selected the vertical center of the visible part of the target as his aim point. In either event the firer is assumed to have selected the horizontal center line for his aim point. Errors in this selection are at least partially included in aim point bias parameters. Having determined the aim point, GEOM assesses whether or not the round missed in elevation. This occurs in three ways. First: the elevation miss distance is positive and larger than the distance from the aim point to the top of the turret. Second: the elevation miss distance is negative and larger in magnitude than the uncovered portion of the target below the aim point. The third way in which a miss in elevation can happen is only possible in the event that the selected aim point (based on amount visible i.e. unconcealed at FIRE) is now covered. Figure 8 shows this case. When this occurs, the elevation miss distance may be positive and still result in a hit on the covered portion of the target i.e. a miss. The second and third cases result in a "sensed" miss. The first case and all misses in deflection result in an "unsensed" miss. If the round did not miss in elevation, deflection is checked. First the aspect angle between the fore and aft center line of the

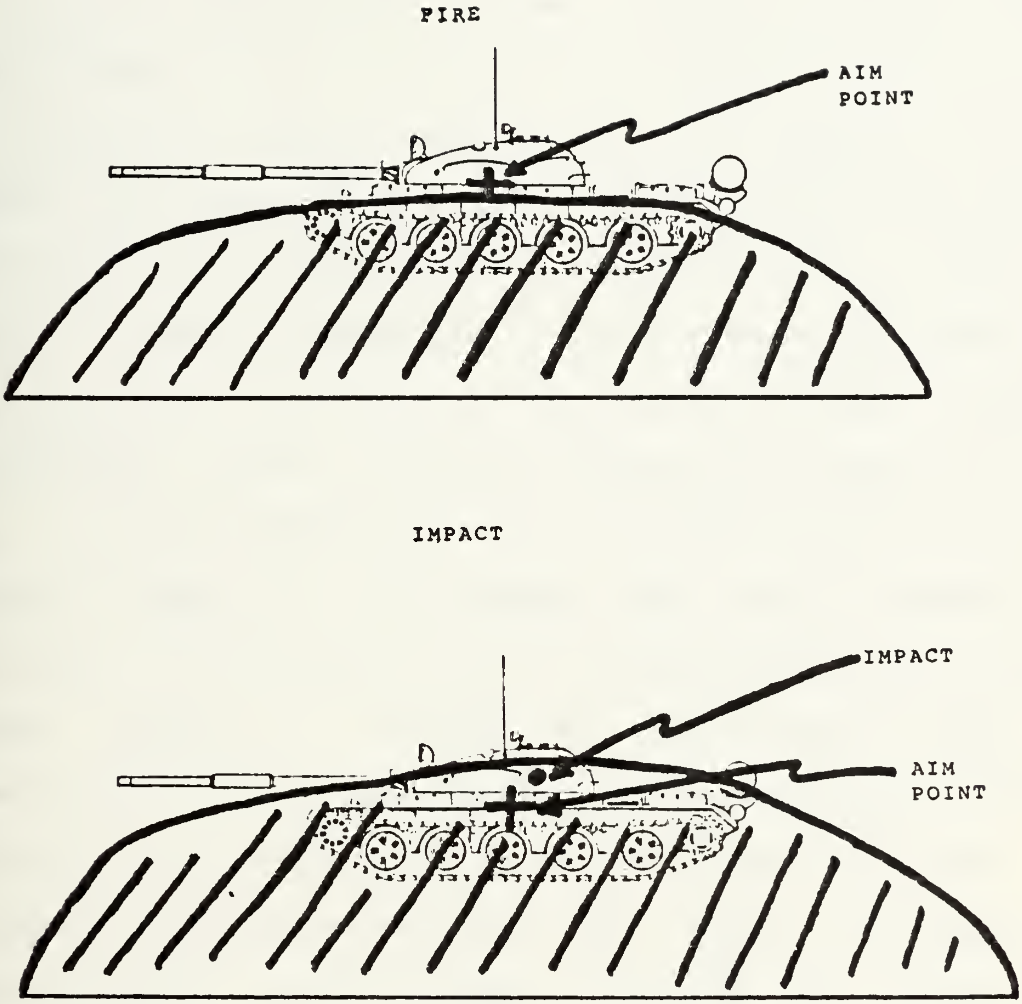
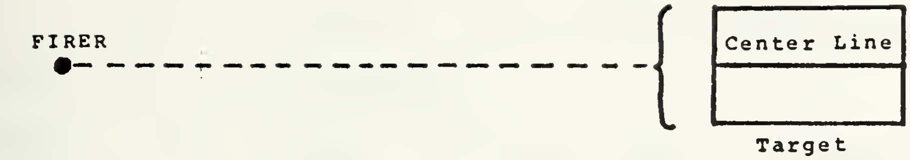


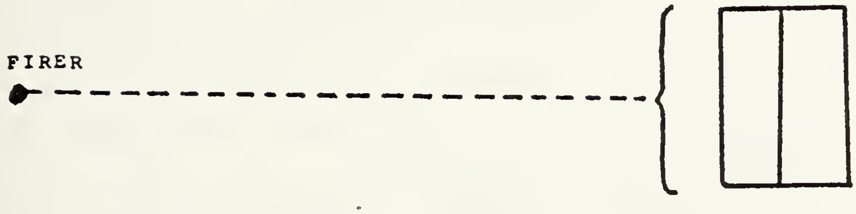
Figure A-8. Aim Point Covered

target and the line of sight from the firer to the target is determined. This angle and the target horizontal dimensions are used to determine the effective width of the target from the point of view of the firer. Figure 9 shows three possibilities.

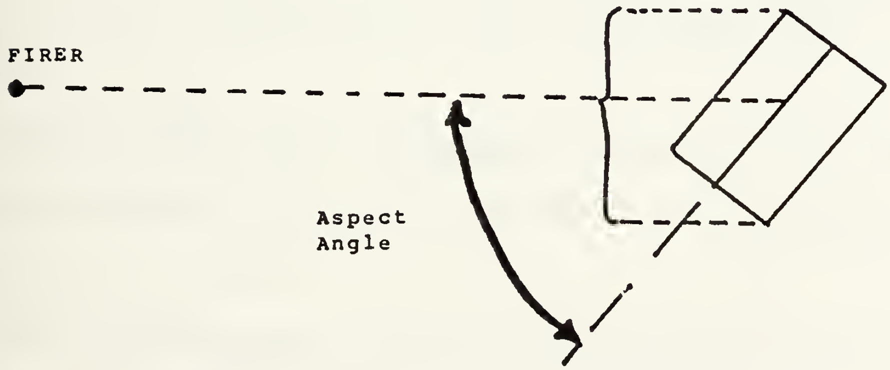
It should be noted that the size and direction of the elevation miss are used to determine whether the horizontal measurements used should be those of the turret or those of the hull. When the effective width has been computed the determination of hit or miss in deflection is made. Since, by assumption the aim point is always the center of the target, if the deflection error is larger than half the effective width of the target, the round missed in deflection. If at any point it is determined that a miss occurred, control is returned to event IMPACT, with an indication that the round missed. If the round did not miss in elevation or deflection, a hit must have occurred. In that case, GEOM determines (by table lookup) the mobility damage function, firepower damage function, and probability of catastrophic kill. These are all numbers between 0 and 1, stored as integers between 0 and 100. The size of the numbers is a function of the type of projectile fired, the type of target, aspect angle, dispersion class and target exposure. Dispersion class is used to account for the fact that a round with a large dispersion is less likely to cause damage than a round with a small dispersion. Target



Effective width is front horizontal dimension



Effective width is side horizontal dimension



Effective width=f(angle,side horizontal width,
front horizontal width)

Figure A-9. Apparent Width of Target

exposure accounts for the difference in vulnerability between a target in hull defilade and one fully exposed. In STAR a target with more than fifty percent of the hull height covered is in "hull defilade" for damage assessment, all others are fully exposed.

When the damage functions and probability of catastrophic kill have been determined, GEOM calls Routine ATRIT. It should be noted that Routine GEOM, with its use of miss distance to determine whether or not a hit occurred, is not used for automatic weapons. No bias and standard deviation data were available for those weapons when STAR was coded. For automatic weapons, COMPUTE calls Routine SUBCAL, which uses probability of hit, and the assumption of round to round independence to determine the number of hits. Based on the number of hits, the damage functions and probability of catastrophic kill are determined and passed to ATRIT.

Routine ATRIT uses the damage functions and probability of catastrophic kill to determine the outcome of the engagement.

STAR distinguishes among four different types of kill, mobility (M kill), firepower (F kill), mobility and firepower (M and F kill), and catastrophic (K kill). A catastrophic kill is immediately apparent to all participants in the battle. Other types of kill are not immediately apparent. A mobility killed entity stops moving.

A firepower killed entity is incapable of firing any armament, but may still move. An entity which is mobility and firepower killed may neither move nor fire.

Entities which are K killed are not engaged again. Rounds fired at the entity are allowed to impact, but no new rounds are fired. Unguided rounds which the K-killed entity fired are unaffected, but command guided rounds such as TOW's, Dragons and Saggers are assessed as misses.

M kills and F kills are not cause for ending engagement, and do not affect the selection of the entity as a target. M and F kills are also not immediately apparent, and the entity may be selected as a target for a period of time which is user input controlled. At the end of that time it becomes apparent that this entity is no longer subject to engagement. The results of the engagement are determined by drawing a random number, uniform on the interval 0 to 1. This one number is used for all comparisons for a single engagement.

First the probability of catastrophic kill is compared to the random number. The catastrophic kill phenomena is assumed independent from round to round. If the random number is smaller, a catastrophic kill is assessed. Otherwise the routine determines if any other type of kill occurred.

Intuitively it is appealing that an entity is easier to M kill or F kill after it has been hit several times, than

on the first hit. The BRL supplied mobility and firepower damage functions model that phenomena. They are not, in general, directly probabilities. They are measures (as we understand it) of the proportion of the entity's remaining mobility and firepower capability that he loses as a result of a hit. Each entity has two attributes which keep track of this loss. M.D is the amount of mobility damage the vehicle has incurred, F.D is the amount of firepower damage. To determine whether an M or F kill has occurred, the new levels of M and F damage are computed by calculating the amount of damage caused by this round, and adding that amount to the existing level. The new level of damage is then compared to the random number. If the damage level is larger than the random number the appropriate type of kill is assessed. If the entity has been both M killed and F killed at the end of the calculations, (either as a result of this engagement or partially as a result of previous engagements) an M and F kill is assessed and a FINAL.DEATH is scheduled. No degradation of the target's ability to move and fire is assessed for levels of M or F damage which do not result in a kill. Table III shows the sequence of calculations to assess kills.

Before control is returned to Event IMPACT, the global variable DAMAGE.NUM is set to a value which indicates the results of the round. Table IV presents those values and their meanings.

Input to ATRIT Target, Firer, EMKILL, EFKILL, Kay Kill

EM Kill: Mobility damage from this round
EF Kill: Firepower damage from this round
Kay Kill: Probability of Catastrophic kill

$$\left.\begin{array}{l} \text{M.D. (Target)} = .5 \\ \text{F.D. (Target)} = .4 \end{array}\right\} \text{Before this round}$$

EM Kill = .60 EF Kill = .70 Kay Kill = .30
Random Draw = .75

Kay Kill is LT .75 No Catastrophic kill

New M.D. = .5+(1-.5).6 = .8

M.D. (Target) = .80 GT .75 Mobility Kill

New F.D. = .4 + (1-.4)(.7) = .4 + .42 = .82

F.D. (Target) = .82 GT .75 Firepower kill

On this round the target was M and F killed.

Note that if the M and F results of previous rounds had not been included in the comparison no kills of any kind would have resulted.

Table III

M and F Kill Computations

Result of Engagement	DAMAGE.NUM
Hit but already MF killed	1
Mobility Damage	2
Firepower Damage	3
Mobility and Firepower Damage	4
Catastrophic Kill	5
Miss	6

Table IV

Upon return to Event IMPACT, the results of the engagement are examined, and the firer takes action as directed by the tactical routines. He may re-engage the same target, select a new target, or move to a new firing position.

As of September 1979, the Field Artillery Module is undergoing substantial revision. The existing module, which has been exercised with the battalion level version of STAR, simulated three fire units, each of four guns, supporting the defense, and ten batteries of howitzers plus one battery of MRL supporting the attack. Three of the cannon batteries supporting the attack were always moving, leaving seven to fire. All attacking (Red) fires were preplanned, except for the MRL battery which because of reload time was only fired

once. All defending fires were target of opportunity, adjust fire missions. Only counter maneuver fires were played.

For the brigade model, significant enhancements are necessary. At least two, and perhaps as many as five battalions must be played on the defending side, and the appropriate number of fire units for a division sized effort must be played on the attacking side. In addition to direct support artillery, reinforcing and general support reinforcing battalions must be simulated.

The forward observer will have the ability to call for fire on preplanned urgent missions, similar to the old Final Protective Fires, on preplanned priority missions, which might be such things as areas for employment of field artillery scatterable mines, precision guided munitions, smoke or perhaps a preplanned concentration on a maneuver choke point. This will simulate missions which have a high priority, but do not have the urgency of an FPF type request. He will also have the ability to fire on targets of opportunity, as he does now.

Additionally, non Forward Observer target acquisition will be modelled. The initial level will include the counter battery and counter mortar radars on the defender's side and equivalent systems for the attacking force. Eventually airborne collectors, both SOTAS and remotely piloted vehicles, as well as direction finding target acquisition

will be included in the model.

The field artillery module must also represent a resource allocation function that determines which requested missions are fired. In the battalion model there were three forward observers (FO) and three fire units. Each FO was allowed to process one mission at a time. As a result, whenever an FO made a request for fire there was a fire unit available to fulfill the request. In the Brigade battle, with many more FO's competing for the resources, and many targets being engaged by multiple fire units, the resource allocation function may be critical in determining whether the field artillery makes a significant contribution or not. The decision making logic will be in the model, but the prioritization and weighting of targets must and will be a user input.

The current module has the ability to play high explosive and dual purpose improved conventional munitions. The Brigade module will include scatterable mines, smoke, and eventually cannon launched guided projectiles (Copperhead).

The method of attack will be predicated on the mission type, i.e. counter maneuver, counter battery, and suppression of enemy air defense, the size of the target, and user input as to how particular targets should be attacked.

The Brigade model will play two sided counter battery

fires, fire units will be caused to displace to alternate firing positions, and ammunition constraints will be played as appropriate. The TACFIRE system will be modelled and a detailed representation of the field artillery radio nets will be included.

The Air-Air Defense Module essentially completes the direct players in the combined arms battle which STAR will model in the near future. These functions are played at the individual entity level. Entities are attack and scout helicopters, fixed wing close air support aircraft, and individual air defense systems such as Stinger, DIVAD, ROLAND and the equivalent OPFOR systems. For those systems where it is appropriate, multiple simultaneous engagements are played. The model simulates visual, infrared and radar detections, laser, wire and command guided munitions and, where appropriate, the command guided to terminal homing switch over.

Aircraft routes from their base to the initial attack positions are preplanned. During engagements the aircraft flight and maneuvers are largely dynamic.

As of September 1979 the Air-Air Defense module is integrated into the ground model and in the process of integration check out runs.

The ultimate goal of STAR, which now appears feasible, is a model which simulates the combined arms battle at the brigade level on realistic terrain using individual tank,

Dragon gunner, field artillery piece, attack helicopter and so on as the primary entities modelled. A great deal of tactical flexibility is provided to the user by his selection of Target Selection tactics, methods of attack, and movement decision and movement coordination logic which are discussed elsewhere. This model will capture the essence of the combined arms battle and the synergistic effects of the interactions of the participants therein.

APPENDIX B. DETAILED METHODOLOGY OF THE DISMOUNTED MODEL

A. GENERAL

This appendix discusses the work which has been completed to date to enhance the STAR ground model to allow for the play of dismounted infantry elements.

B. METHODOLOGY

1. Entity Representation

The dismounted version of STAR has increased the number of entity types. Currently the following dismounted elements are simulated in the model.

a. M16 Rifleman

b. M60 Machinegunner

c. SAW Machinegunner

d. M203 Grenadier

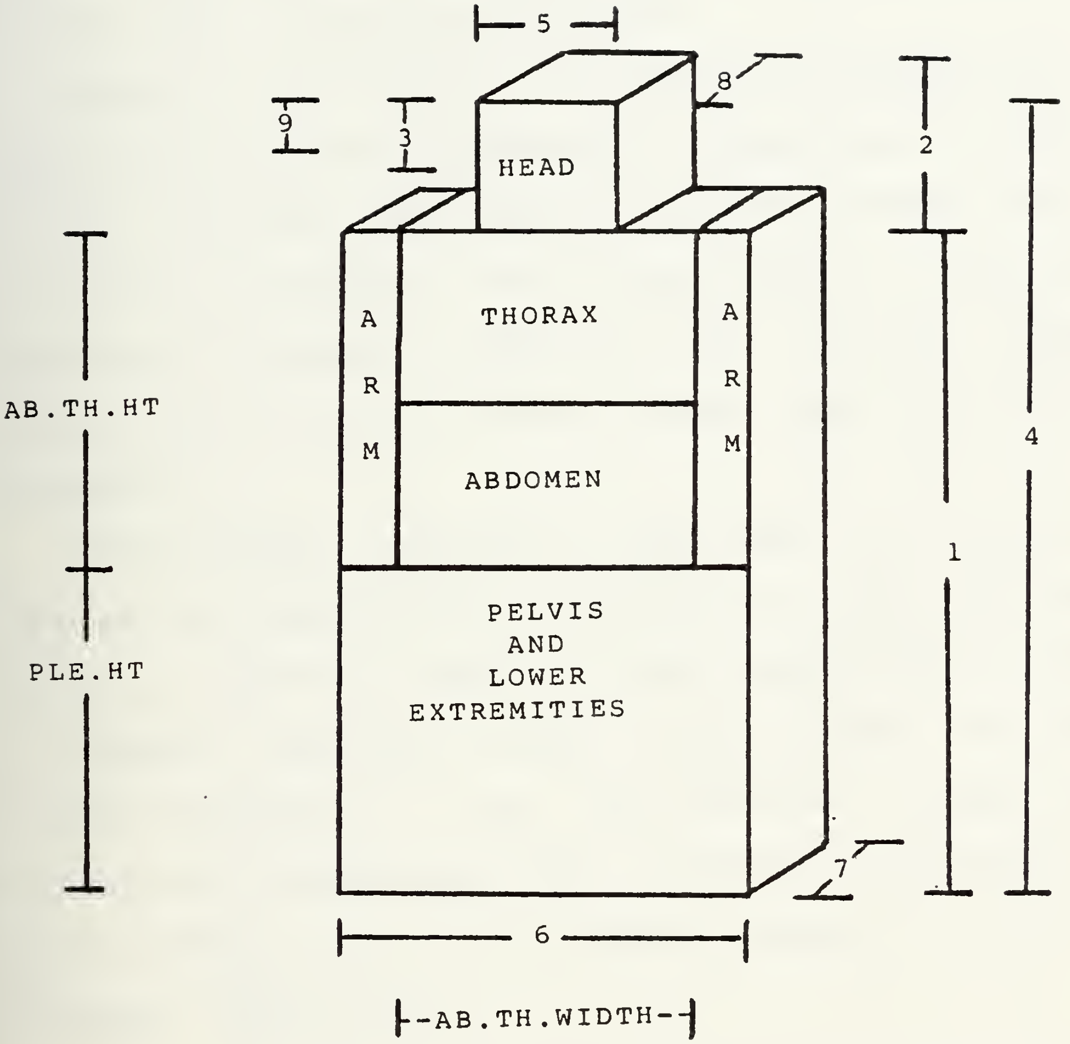
e. Dragon Gunner

f. RPG Gunner

g. AKMS Rifleman

h. PKM Machinegunner

The physical representation of the entity is shown in Figure B-1. The arabic numerals refer to tiers of the TARDIM array. This array is a three dimensional array indexed by the system type of the element (SYS.TYPE), the weapon type of the element (WPN.TYPE), and the desired dimension (the arabic numeral in the figure). Additionally, three global



NOTE: THE ARABIC NUMERALS REFFR TO ELEMENTS OF THE TARDIM ARRAY.

Figure B-1

variables contain the dimensions of the abdomen and thorax height (AB.TH.HT), the width of the abdomen or the thorax (AB.TH.WIDTH), and the height of the pelvic and lower extremities (PLE.HT). These variables are used to determine the body part hit by the projectile.

Each of the elements is classified according to his system type and weapon type and is to some degree classified by the particular ammo type he is firing. Table B-1 lists the current structure in the ground model. In addition to providing the amount of ammunition left in a given category, the ammo type acts as a flag in target and round selection procedures.

The tactical structure in the model is virtually the same as the real tactical structure of a unit. The element acts as an individual entity in the simulation for purposes of movement, detection, target selection and other entity related activities. He can also belong to a squad set, a platoon set, a company set and is linked to a battalion set and a brigade set by his company commander and battalion commander respectively.

2. Detection

In the current version of STAR the driving routine is Routine STEP.TIME. This routine schedules a detection to occur in the future if the appropriate conditions exist. If a detection is made, a check is made to determine which if

AMMUNITION TYPES

WPN.TYPE	SYSTEM NAME	AMMO1	AMMO2	AMMO3	AMMO4	AMMO5	AMMO6
6	DRAGON	NA	DRAGON	M16	NA	NA	NA
10	M16 RIFLEMAN	LAW	VIPER	M16 AR	M16	M18 CLAYMORE	HAND GRENADE
11	M60/SAW MACHINEGUN	LAW	VIPER	M60	SAW	M18 CLAYMORE	HAND GRENADE
12	M203 GRENADIER	LAW	VIPER	M203HE	M203DP	M18 CLAYMORE	HAND GRENADE
13	RPG GUNNER	AKMS	NA	RPG	NA	NA	HAND GRENADE
14	AKMS RIFLEMAN	NA	NA	AKMS	NA	NA	HAND GRENADE
15	PKM MACHINEGUN	NA	NA	PKM	NA	NA	HAND GRENADE

Table B-1

any of the detected targets is to be engaged. If a target is fired upon, subsequent calls are made to events FIRE and IMPACT to assess the results. For vehicles Routines CARDIO and SIGHT are used to deal with detections. In the case when a dismounted infantry element is being detected, however, Routine STEP.TIME calls Routine DISMTD.CARDIO to calculate the detection time.

The initial checks in Routine DISMTD.CARDIO are for the boundary cases. If the target is within 10 meters a detection time of one second is assumed and the target is scheduled to be detected in that amount of time. If the range to the target exceeds 1200 meters, the detection time is set to 99 seconds and control returns to the calling routine. The probability of detection at this range as computed by the model is very close to zero; therefore the 1200 meter check is simply to save computer time. The 99 second detection does not occur if the detection time exceeds DELTA.T (currently 30 seconds). If the detection time is determined to be less than DELTA.T, a detection is scheduled for the firer. The first check made is to determine if the observer is looking in the proper direction. The observer is assumed to be searching for targets in a particular fan of a cardioid distribution. The observer's area of search is variable. If he has not detected any targets, the observer will expand his area of search up to 360 degrees. When a target is detected, the

search sector is reduced to 30 degrees. At the time that a detection is to be checked, the cardioid is oriented in the firer's primary search direction. The probability that the observer was looking in the sector where the target is, is then computed. The derived probability is then compared to a random number, and a determination is made as to whether the target was detected. If the random number exceeds the probability that the observer was viewing the appropriate sector (P.SUB.K), then the detection time is again set to 99 seconds and the program returns to the calling routine.

If the observer detects a target, then the detection time (DET.TIME) must be determined. The time to detection model is the ASARS detection model. Three values are needed to calculate the time to detection. They are:

a. The terrain effect

b. The range effect

c. The speed effect

The terrain effect is determined from the complexity of the terrain in which the target is located. If the complexity is high, the value of the TC.FACTOR is 1.35. If the complexity is medium, the value of TC.FACTOR is 1.49 and if the complexity is low, the TC.FACTOR is 0.53. The interpretation of high, medium and low complexity can be found in ASARS BATTLE MODEL, Book 2, Volume II-A, Narrative Description.

The second effect in determining the mean time to detection is the crossing velocity factor. Once the crossing

velocity (X.VELOCITY) of the target is determined, the crossing velocity factor is computed with the following equation:

CROSSING.VEL.FACTOR=1.39-0.76*(X.VELOCITY)

The final factor is the range factor which is determined by means of the following equation:

RGE.FACTOR=0.018+(0.0058*(R/PER.FULL.EXPO))

PER.FULL.EXPO is the percent of the target which is fully exposed. R is the range to the target. Both equations are the result of linear regression applied to the HumRRO TAS data as explained in the ASARS reference listed above. The mean detection time (M.D.T) is now defined by the following expression:

M.D.T=1.1*EXP.F(TC.FACTOR+CROSSING.VEL.FACTOR+RGE.FACTOR)

EXP.F is a SIMSCRIPT exponential function. The variable M.D.T appears in the routine as MEAN.DETECT.TIME.

The TAS data was initially tested to determine if the detection phenomena for dismounted elements was exponentially distributed. Test results rejected the hypothesis and a subsequent test was made to determine if a

log-normal distribution better described the process. This hypothesis was accepted by ASARS and is utilized here. The mean of the log-normal detection times is MEAN.DET.TIME as derived above. The fitting of the data to the log-normal distribution resulted in a residual mean square error of 0.81. This number is an unbiased estimator of the standard deviation of log-normally distributed detection times and is used. A random draw is now made from the log-normal distribution with the parameters described. The result is the time required for the observer to detect the target by the firer. If this time is smaller than DELTA.T then the detection will take place if the proper conditions (e.g. line of sight) still exist at the detection time.

3. Target Selection

The initial input in the target selection process is the target selection tactic. Any one of fourteen tactics may be selected by the user for each system type and weapon type combination. These fourteen tactics are described in TACTICAL PARAMETERS AND INPUT REQUIREMENTS FOR THE GROUND COMPONENT OF THE STAR COMBAT MODEL, Dr. Samuel H. Parry and Lieutenant Colonel Edward P. Kelleher, October 1979. They are:

(1) Attempt to acquire your platoon leader's target. Failing this search your platoon to determine which of your targets are not being engaged by another platoon member. From this reduced set, engage your highest priority target using the alternate ammunition type

that you specified in array POINT.HOLD.

(2) Attempt to acquire your platoon leader's target. Failing this, search your platoon to determine which of your targets are not being engaged by another platoon member. From this reduced set, engage your highest priority target with the ammunition specified in the target selection array (the ARRAY for this system/weapon combination).

(3) Attempt to acquire your platoon leader's target. Failing this, search your platoon to determine which of your targets are not being engaged by another platoon member. From this reduced set, engage your highest priority target. If all targets are engaged do not engage any targets.

(4) Same as 1, except the company is searched.

(5) Same as 2, except the company is searched.

(6) Same as 3, except the company is searched.

(7) Attempt to acquire your platoon leader's target. Failing this, engage your highest priority target regardless of its engagement status.

(8-14) These tactics are identical to 1 through 7, except that no attempt is made to acquire the platoon leader's target.

Thus, crewdrills 1 through 7 attempt a platoon leader's handoff as the first choice for a target selection tactic. Crewdrills 8 through 14 move directly to an evaluation of each target in the selector's list of detected targets.

If a platoon handoff is not accomplished (because it failed or was not specified) then the following statements apply:

(1) Crewdrill 1,4,8, and 11 are "missile savers" in that if all targets are being engaged, an alternate ammunition type (usually some sort of automatic weapon ammunition) will be specified to engage the target.

(2) Crewdrills 2,5,9, and 12 will always result in the

selection of a target if a target is available and engageable.

(3) Crewdrills 3,6,10, and 13 will only allow selection of an unengaged target.

(4) Crewdrills 7 and 14 will always select the highest priority target from a selector's list of detected targets, regardless of that target's current engagement status.

A typical input sequence which portrays the target selection input process can be found at Table 4-2.

A closer look at the POINT.HOLD array (Table B-1a) and the ARRAY array (Table B-1b) points out the functionality and the flexibility of the target selection process. The first line of POINT.HOLD is:

3	10	14	1500	250	350	350	350	300	350	991	991	0
a	b	c	d		e				f			g

(a) The system type (SYS.TYPE) of the system.

(b) The weapon type (WPN.TYPE) of the system.

(c) The crew drill tactic to be employed. There are 14 possible selections as explained above.

(d) The maximum acquisition range of the system and the weapon type combination, in meters.

(e) The maximum opening ranges in meters for AMMO1, AMMO2,AMMO3, and AMMO4 for this system and weapon combination.

(f) The muzzle velocities in meters per second for AMMO1, AMMO2, AMMO3, and AMMO4 for this system and weapon code combination.

POINT.HOLD

3 10 14 1500 250 350 350 350 300 350 991 991 0
2 [illegible] 3

ARRAY

3	3 14 1 3	3 15 2 3.	3 13 3 3
3	2 8 11 1	3 14 12 3	3 15 13 3
3	2 8 21 1	3 14 22 3	3 15 23 3
3	2 8 31 1	3 14 32 3	3 15 33 3
3	2 8 41 1	3 14 42 3	3 15 43 3
3	3 14 51 3	3 15 52 3	3 13 53 3
3	3 14 61 3	3 15 62 3	3 13 63 3
[illegible]	0 0 0 0		
1	0 0 0 0		
[illegible]	0 0 0 0		
	[illegible] 0 0 0		
1	0 0 0 0		
[illegible]	0 0 0 0		

Table B-2

(g) Indicates whether or not all ammunition types may be fired on the move.

1-Yes

0-No

The next line of input appears as follows:

2	2	3	3
h	i	j	k

(h) WE.HIT TACTIC number to be used.

(1) Go to full defilade if number of hits equals WH.1 or number of hits equals WH.4 and number of misses equals WH.3.

(2) Reselect another target without going to full defilade.

(3) Go to full defilade if the number of rounds fired since last defilade is larger than WH.5.

(i) WE.MISS TACTIC number to be used.

(1) Go to full defilade if the number of misses equals WM.1 or number of hits equals WM.2.

(2) Reselect another target without going to full defilade.

(3) Go to full defilade if the total number of shots is greater than WM.3 following a miss.

(j) Indicates the time in seconds to remain in

full defilade after a WE.HIT, WE.MISS sequence.

(k) Indicates an alternate ammunition type which may be fired on the move.

The next sequence for the dismounted version of STAR is the thirteen layer array referred to as ARRAY. Each tier of this array indicates a prioritization of targets for a given range band. The tier numbers and corresponding range bands are:

TIER	RANGE
1	0-49
2	50-99
3	100-149
4	150-199
5	200-249
6	250-299
7	300-349
8	350-399
9	400-449
10	450-499
11	500-999
12	1000-1999
13	2000+

A typical tier level might appear as follows:

3	2 8 1 1	3 14 2 4	3 15 3 4
a	b	c	d

(a) There will be 3 sets of 4 entries in this tier of ARRAY.

(b) 2 is the system type of the target.

8 is the weapon type of the target.

1 is the priority of the target.

1 is the type of ammo to be fired at this priority target if it is available.

(c) An AKMS rifleman is the second priority target if ammo type 4 is available.

(d) A PKM machinegunner is the third priority target also using ammo type four.

The flexibility of this system lies in the ability to specify any target on the battlefield for any priority of desired engagement within a range band and to allow the selection of different ammunition types according to the target type and range to the target.

4. Fire and Casualty Assessment

The method of casualty assessment depends on the type of projectile used. If the projectile is a LAW, VIPER, M203DP, or RPG, the data available are the probability of mobility kill, firepower kill, catastrophic kill and expected casualities. Routine INF.COMPUTE determines the needed indices for range, orientation angle, velocity, target type, and type of round fired. These are then used to look up the kill probabilities in the appropriate lethality arrays. The kill probabilities are then passed into routine ATRIT for evaluation. Since this process is very similiar to the lethality play currently in STAR, it will not be discussed further here.

For small arms non-fragmenting rounds fired against exposed personnel targets a different procedure is used.

First the target and weapon and the appropriate indices for accessing the lethality data are determined. The factors affecting the outcome are:

a. Range to the target

b. The defilade status of the target

c. The number of rounds fired

d. The aspect angle of the target

These factors are used to obtain the following information:

a. Weapon aim error

b. Round to round ballistic error

c. Horizontal distance from the point of impact of the first round to the center of impact of the subsequent rounds.

d. Vertical distance from the point of impact of the first round to the center of impact of the subsequent rounds.

e. The standard deviation of the horizontal distance from the point of impact of the first round to the center of impact of the subsequent rounds.

f. The standard deviation of the vertical distance from the point of impact of the first round to the center of impact of the subsequent rounds.

g. The standard deviation of the deflection error of a subsequent round about the center of impact of the subsequent rounds.

h. The standard deviation of the elevation error of a subsequent round about the center of impact of subsequent rounds.

The use of this data in DISMOUNTED-STAR is explained below.

Initially it is assumed that the firer of a weapon having a non-fragmenting small arms projectile (M16,AKM,M60,SAW) fires at the center of mass of the visible (exposed) portion of the target. As a result of an improperly zeroed weapon, an incorrect aiming technique, or other source, a firer may have aim error. This error is a common error and is stored in array AIMERROR. Data from this array is accessed and assigned to the variable AIMERR. This error, called round to round ballistical dispersion, is represented by the variable BALLERR. Even if the firer had consistently perfect aim, rounds he fired would impact at different points due to the ballistic differences of the rounds. Causes of ballistical error include projectile weight, wind effects, and varying amounts of propellant. The total first round miss distance is then represented by the following equation:

$$\sigma_X = \sigma_Y = \sigma = \sqrt{(\mathrm{AIMERR})^2 + (\mathrm{BALLERR})^2}$$

To provide the proper dispersion, these values are

multiplied by a random number from a Normal (0,1) distribution. This is, of course, equivalent to drawing the horizontal and vertical dispersions from the following distribution:

$$N(0,\sigma_X) = N(0,\sigma_Y) = N(0,\sigma)$$

For burst-fire weapons, a three distribution case is utilized. Figure B-2 delineates the elements of error for these weapons. The procedure for locating the impact miss distances of the first round from the aim point has been discussed, therefore this section will commence with the impact of the first round. Once the first round of a burst has impacted, subsequent rounds are modelled as landing about the center of impact of those subsequent rounds. Associated with this center of impact are the following:

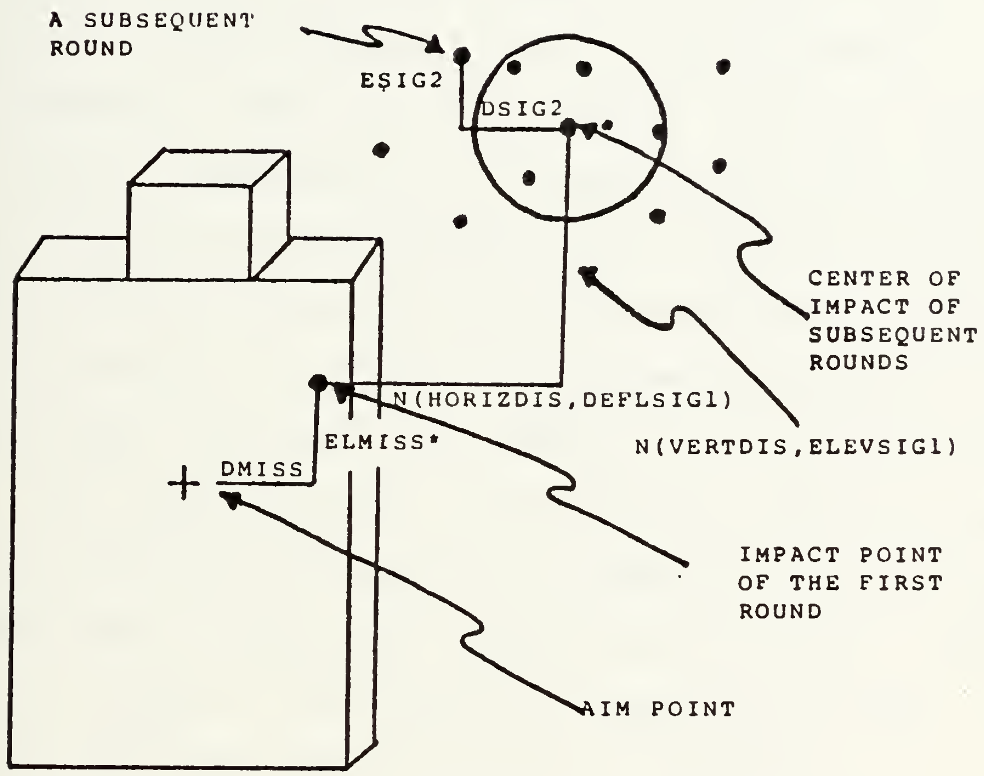
a. The horizontal distance from the impact of the first round to the center of impact of the subsequent rounds. This is represented by the variable HORIZDIS.

b. The standard deviation associated with HORIZDIS which is defined as DEFLSIG1.

c. The vertical distance from the impact point of the first round to the center of impact of the subsequent rounds. This is represented by the variable VERTDIS.

d. The standard deviation associated with VERTDIS,

VARIABLES ASSOCIATED WITH BURST FIRE WEAPONS



NOTE: THE VALUE OF ELMISS IS CONVERTED TO METERS IN THE LATTER PART OF THE ROUTINE. EMISS THEN MAINTAINS THE INITIAL ELEVATION MISS IN METERS WHILE ELMISS IS USED TO TRACK SUBSEQUENT ROUNDS.

Figure B-2

which is defined as ELEVSIG1.

In locating the center of impact the following procedure is used to draw the deflection distance from a NORMAL(HORIZDIS,DEFSIG1) distribution and to draw the elevation distance from a NORMAL(VERTDIS,ELEVSIG1) distribution. This is accomplished in both cases by adding HORIZDIS to a N(0,1)*DEFLSIG1 and by adding VERTDIS to a N(0,1)*ELEVSIG1. Once this offset from the first round's impact point is calculated it is added to the first round's offset resulting in a total offset from the aimpoint to the center of impact. The total offset for deflection is CIPDEFL and for elevation is CIPEL. Subsequent projectiles are dispersed about this center of impact with two contributing errors. The first of these is individual round to round dispersion characteristics while the second is an associated subsequent round dispersion for deflection (DSIG2) and for elevation (ESIG2). The offset is measured in mils and calculated from the center of impact as follows:

$$\text{SIGMAX} = \sqrt{(\text{DSIG2})^2 + (\text{BALLERR})^2}$$

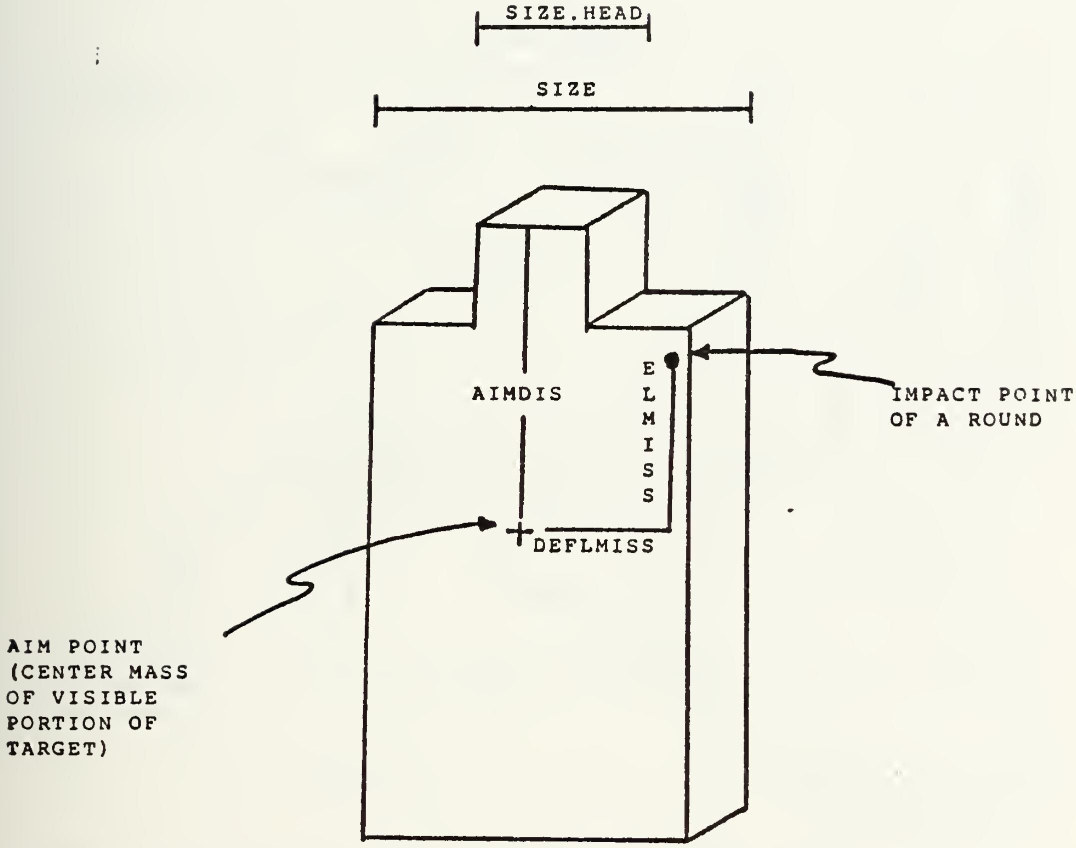
$$\text{SIGMAY} = \sqrt{(\text{ESIG2})^2 + (\text{BALLERR})^2}$$

The variables SIGMAX and SIGMAY do not actually appear in

any of the routines. Instead the values for each are added directly to CIPDEFL and CIPEL resulting in a new DEFLMISS and ELMISS for each round. That is, the total deflection and elevation miss distances from the aimpoint are calculated for each projectile individually. Once the horizontal and vertical dispersions are determined in mils, they are converted to meters. The variable DEFLMISS now represents the magnitude of the total deflection miss distance in meters. ELMISS represents the total elevation miss distance in meters. A third variable, AIMDIS, is now introduced as the positive distance down from the top of the target the firer aimed. The apparent width of the body (SIZE) and the apparent width of the head (SIZE.HEAD) are now computed. Additionally the necessary target dimensions are accessed from the TARDIM array. Figure B-3 depicts a target entity with the variables described above.

The next step in the casualty assessment process is to determine if the shot went over the target's head. Figure B-4 illustrates this possibility. If the elevation miss distance (ELMISS) is greater than the distance down from the top of the target the firer aimed (AIMDIS), then the projectile was above the head of the target. A miss is indicated and the next round is assessed. If the round did not go over the target, two checks are made to see if the round went low into the dirt. Figure B-5 depicts the first

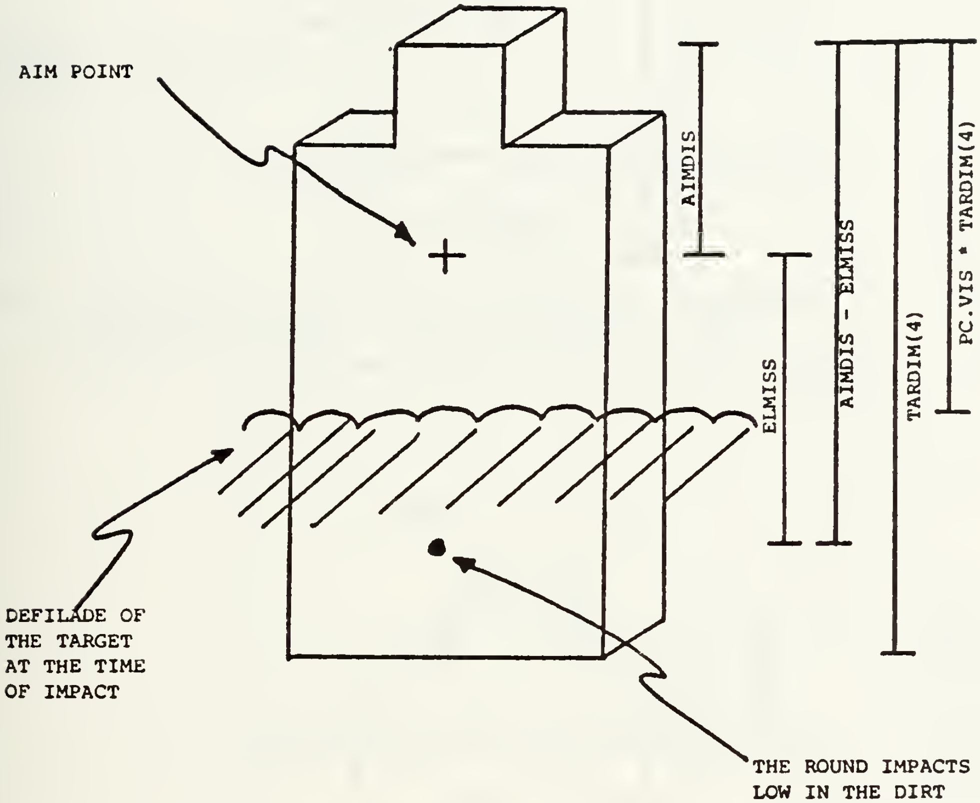
FIRST ROUND ERRORS



NOTE: AIMDIS IS THE DISTANCE DOWN FROM THE TOP OF THE TARGET THE FIRER AIMED.
DEFLMISS IS THE MAGNITUDE IN METERS OF THE TOTAL DEFLECTION MISS DISTANCE.
ELMISS IS THE TOTAL ELEVATION MISS DISTANCE IN METERS.

Figure B-3

MISS OVER TOP OF TARGET

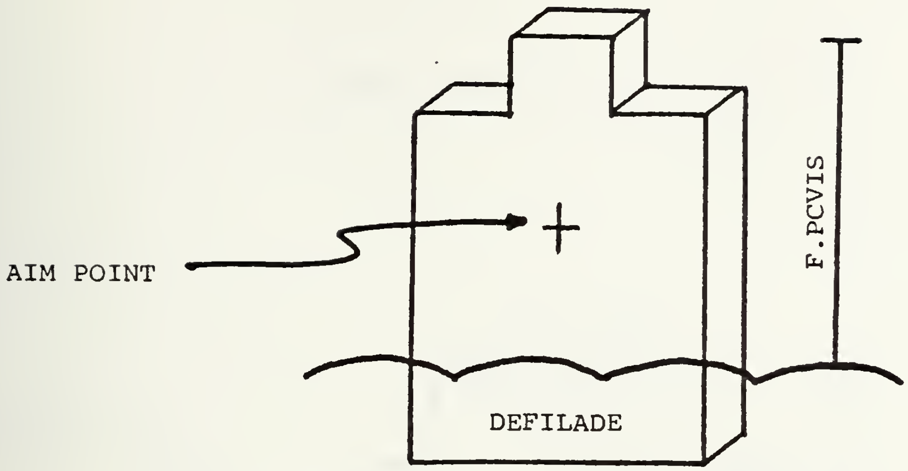


NOTE: IF THE ROUND IMPACTED LOWER THAN THE PERCENT VISIBLE OF THE TARGET AT THE TIME OF IMPACT IT LANDS LOW IN THE DIRT.

Figure B-4

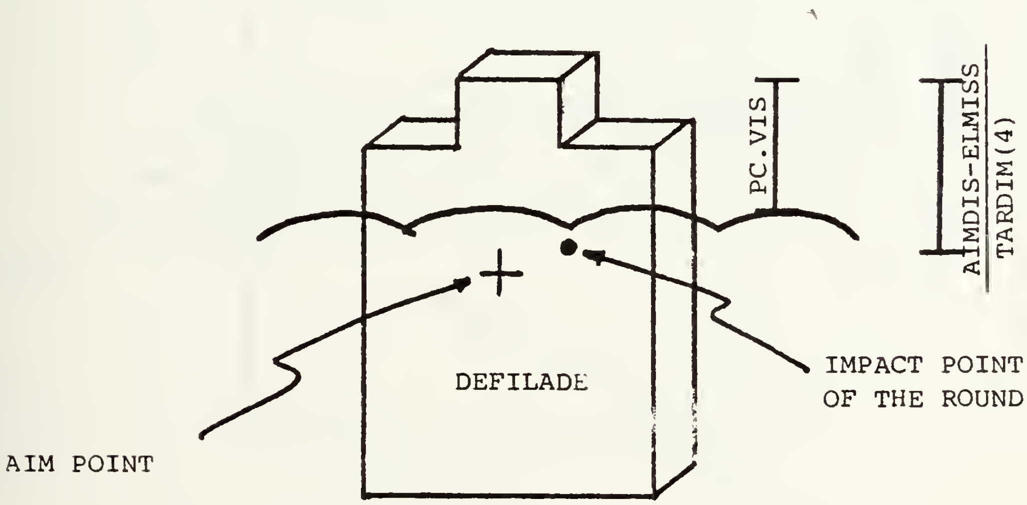
MISS LOW IN THE DIRT

TARGET AT FIRE



NOTE: AT THE TIME OF FIRE, THE PERCENT OF THE TARGET VISIBLE IS F.PCVIS. THE AIM POINT IS THEN PLACED AT THE CENTER MASS OF THE VISIBLE TARGET.

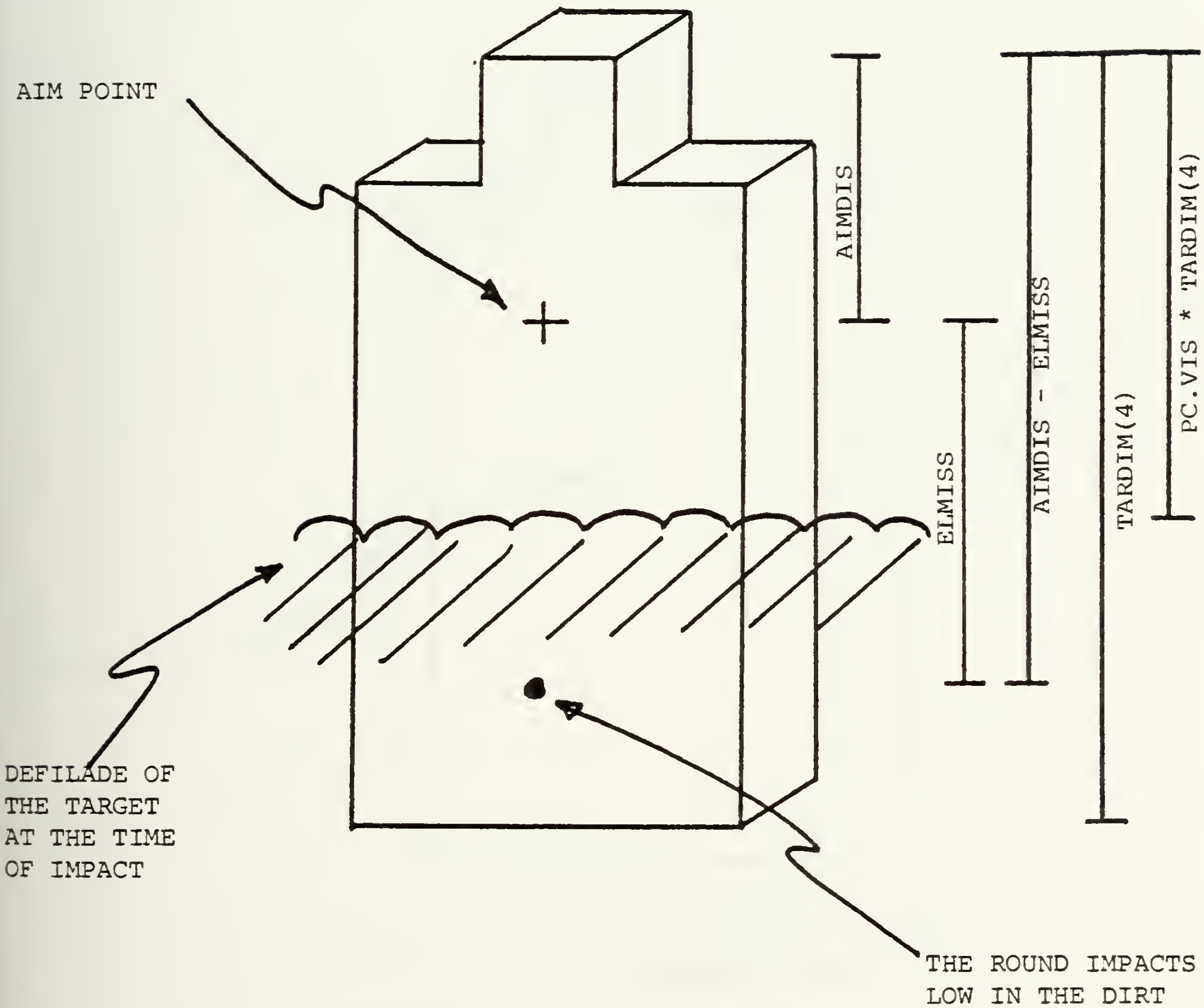
TARGET AT IMPACT



NOTE: Because the target was moving it is possible that the percent visible of the target at impact (PC.VIS) has decreased from the time of fire. If PC.VIS is less than the distance from the top of the target (expressed as a per cent) the round has impacted ((AIMDIS-ELMISS)/TARDIM(4)), the round hit low in the dirt

Figure B-5

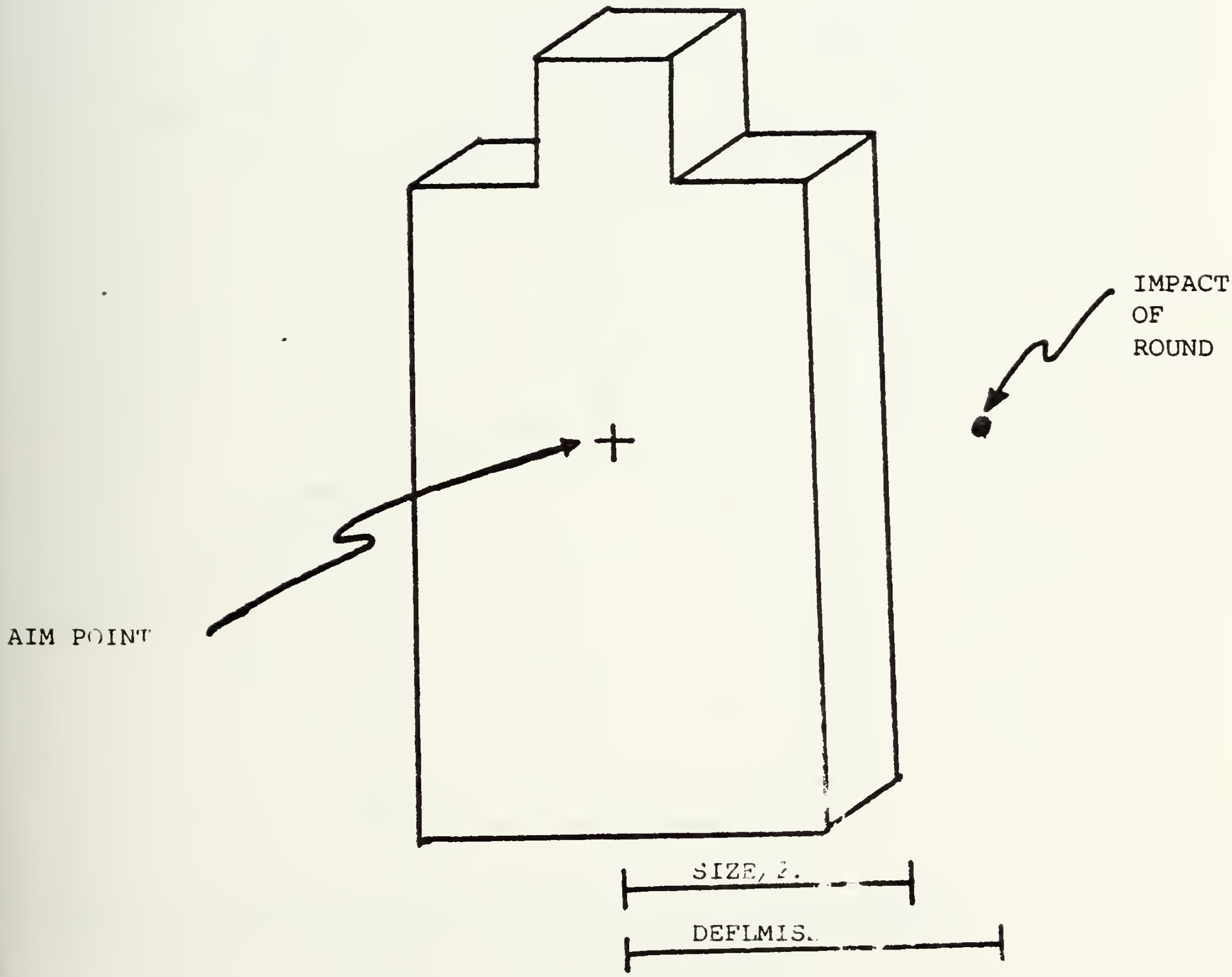
CASE 2 MISS LOW IN THE DIRT



NOTE: IF THE ROUND IMPACTED LOWER THAN THE PERCENT VISIBLE OF THE TARGET AT THE TIME OF IMPACT, IT LANDS LOW IN THE DIRT.

Figure B-6

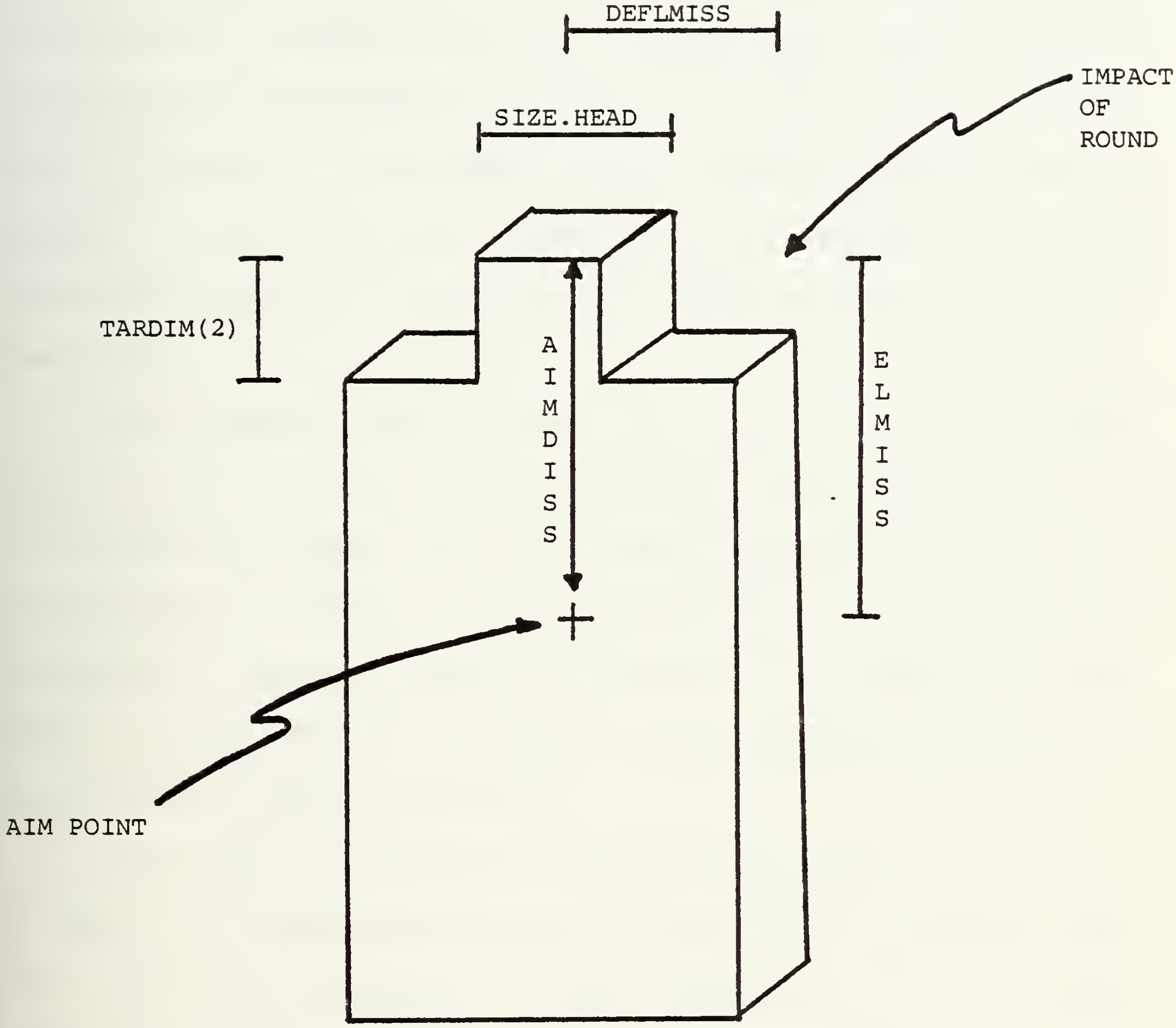
MISS DUE TO DEFLECTION ERROR



NOTE: IF THE MAGNITUDE OF DEFLMISS EXCEEDS THE APPARENT WIDTH OF THE TARGET DIVIDED BY TWO, THE TARGET WAS NOT HIT.

Figure B-7

MISS WIDE OF HEAD OVER THE SHOULDER



NOTE: ONCE IT HAS BEEN DETERMINED THAT THE ROUND IS BELOW THE HEIGHT OF THE TOP OF THE HEAD, A CHECK IS MADE AS FOLLOWS. IF ELMISS IS GREATER THAN ZERO AND AIMDIS-TARDIM(2)-ELMISS IS LESS THAN ZERO, THE ROUND IS SOMEWHERE ABOVE THE SHOULDER. NOW IF DEFLMISS IS GREATER THAN SIZE.HEAD/2., THE HEAD WAS ALSO MISSED.

Figure B-8

of these checks. The first check is for an elevation miss (ELMISS) greater than zero causing the round to impact above the aim point. If the percent visible of the target at the time of impact (PC.VIS) is less than the percent of the target height above the point at which the round impacted ((AIMDIS-ELMISS)/TARDIM(2)) then the round landed in the dirt in front of the target. This situation occurs when a target is moving behind cover and the amount of cover increases between the time of fire and the time of impact of the round.

The second check is for an ELMISS less than zero causing the round to impact below the aim point. This is illustrated in Figure B-6. The check determines whether the distance down from the top of the target that the round impacted is greater than the distance from the top of the target to the top of the cover. If the distance is greater, the round hit low in the dirt.

If no miss in elevation is assessed, the next step is to check for deflection error. Figure B-7 illustrates the first of these checks. If the magnitude of the total deflection miss distance (DEFLMISS) is greater than half the apparent width of the target(SIZE/2), then the round missed the target.

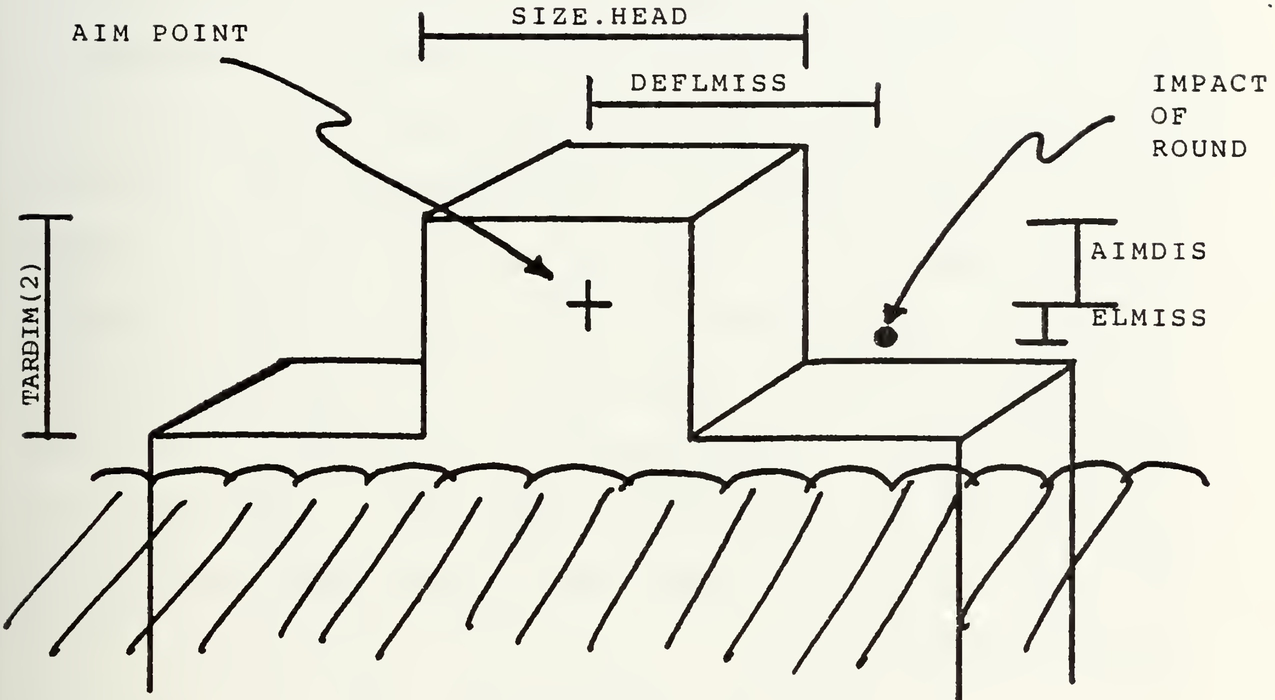
The only remaining potential miss areas are those above the shoulders yet wide of the head. Figure B-8 depicts the

first of two cases which could cause a miss. This occurs when the firer was aiming at the body, but the round went higher than shoulder level. If AIMDIS-TARDIM(2)-ELMISS is less than zero then the round is above the shoulder. The round has already been determined to be below the top of the head by a previous check. Now if the total deflection miss distance (DEFLMISS) is greater than half the apparent width of the head (SIZE.HEAD/2), then the round missed the head over the shoulder. The second situation is concerned with the case where the aimpoint was on the head of the target. Figure B-9 illustrates this case. If the DEFLMISS is greater than SIZE.HEAD/2 the shot is again over the shoulder.

The next stage of the casualty assessment process depends on a user input option variable. INF.CAS.ASSMT indicates whether or not the user desires detailed casualty assessment play. A value of zero indicates the user desires all hits be recorded as kills. That is, if a target is hit by a round then it is incapacitated immediately regardless of the body part struck by the projectile. In this case routine FINAL.DEATH is scheduled immediately. If the value of INF.CAS.ASSMT is equal to one, the user desires detailed casualty assessment play. This is accomplished by a call to routine INCAPACITATE.

Routine INCAPACITATE was derived from the ASARS Battle Model, but has been modified to some extent for use in STAR.

CASE 2 MISS WIDE OF HEAD OVER THE SHOULDER



NOTE: IN THIS CASE THE AIM POINT WAS ON THE HEAD BUT THE SHOT WAS LOW. IF THE SHOT WAS ABOVE THE SHOULDER, A CHECK IS MADE TO SEE IF IT HIT THE HEAD. THIS PROCEDURE IS DONE AS FOLLOWS. IF AIMDIS-ELMISS IS LESS THAN TARDIM(2), THE BULLET IS SOMEWHERE ABOVE THE SHOULDER. NOW IF DEFLMISS IS GREATER THAN SIZE.HEAD/2., THE HEAD IS MISSED. THIS ENTIRE CHECK DEPENDS ON THE PREVIOUS CHECK THAT THE ROUND HAS IMPACTED NO HIGHER THAN THE TOP OF THE HEAD.

Figure B-9

This routine begins by determining the body part struck by the round, the type of round, and the range to the target. This information is then used to access three coefficients of the equation:

Y=EXP(a+b*R+c*R*R)

where Y is the probability of incapacitation by a given time (30 seconds, 5 minutes, or 30 minutes) and R is the range from the firer to the target. The first set of coefficients drawn are for the probability of incapacitation within 30 seconds of being hit by the round. The probability of incapacitation is assigned to the appropriate attribute of the entity. There is an attribute for each body part to store this probability of incapacitation. These attributes are P.HD.INCAP, P.TH.INCAP, P.AB.INCAP, P.ARM.INCAP and P.PLE.INCAP. Additionally, the number of hits to each body part is stored for the entity as one of five attributes. These are N.HD, N.TH, N.AB, N.ARM and N.PLE.

These values are stored until routine BRST.FIRE sets the value of CALC to one which indicates that the last round in a burst has been fired on the target and the total body probability of incapacitation should now be calculated. The equation for the total body probability of incapacitation is:

P.BODY.INCAP=1-((1-P.HD.INCAP)**N.HD)*((1-P.TH.INCAP)**N.TH)
*((1-P.AB.INCAP)**N.AB)*((1-P.ARM.INCAP)**N.ARM)

*((1-P.PLE.INCAP)**N.PLE)

The variables in the above equation are defined as follows:

P.BODY.INCAP - The probability of total body incapacitation.

P.HD.INCAP - The probability of incapacitation given the element was hit in the head.

P.TH.INCAP - The probability of incapacitation given the element was hit in the thorax.

P.AB.INCAP - The probability of incapacitation given the element was hit in the abdomen.

P.ARM.INCAP - The probability of incapacitation given the element was hit in the upper extremities.

P.PLE.INCAP - The probability of incapacitation given the element was hit in the pelvis or lower extremities.

Once the total body probability of incapacitation has been calculated a uniform random number is drawn on the interval (0,1) and compared. If the random number is greater than the probability of incapacitation, the target element has not suffered incapacitation within the first 30 seconds. In this case the probability of the total body incapacitation by five minutes is then calculated. The random number test is again applied. If it fails the same procedure is followed to test incapacitation by thirty minutes. If the soldier is not determined to be incapacitated by thirty minutes, he carries

forward a residual potential for incapacitation on subsequent rounds. The target's number of hits attribute for each body part is retained and used as an exponent in the next calculation of incapacitation probability for subsequent rounds.

If at a given stage, it is determined that the entity is incapacitated, the exact degree remains to be determined. The first step in this process is to classify the target element into one of the following four categories:

1. The target is hit only in the arms or the target is in a defensive mode and is hit in the head, thorax, abdomen, or the upper and lower extremities.

2. The target is in a defensive posture and is hit only in the legs.

3. The target is in an assault posture and has wounds anywhere on his body but in the legs only or in the arms.

4. The target is in an assault posture and is hit only in the legs.

The program continues by calculating two proportionality factors TF and TO. The detailed derivation is outlined in the ASARS documentation. These factors are:

TF=0.16*((PA-0.5)**2)

TO=1-0.2*PA

where PA is the probability of total body incapacitation if the target element was assaulting.

If the target element is determined to be in category 1, the target is assumed to be at least a firepower casualty. A comparison is then made to a random number to determine if the target suffered only a firepower kill (P.ONLY). If the target suffers only a firepower kill routine ATRIT is called and the incapacitation is scheduled in TIME units. TIME can fall into one of three ranges:

1. Uniform (0,30) seconds.
2. Uniform (30,300) seconds.
3. Uniform (300,1800) seconds.

The range used is dependent on the time by which the target is determined to be incapacitated. If the target suffers more than a firepower kill, the following probabilities are calculated:

P.MFKILL=TF*(1-TO)

P.KKILL=TF*TO

Routine ATRIT is then called to determine the appropriate level of incapacitation, which is then scheduled in TIME units.

If the target element is determined to be in category 2, the target is assumed to be at least a mobility and firepower kill. The probability that the target suffers only this level of incapacitation is P.ONLY=1-TO. The random number check is then made; if the random number is larger, routine ATRIT is called and a MFKILL is scheduled in TIME

seconds. If the random number is less than or equal to P.ONLY, the probability of catastrophic kill is determined as P.KILL=TO. Routine ATRIT is called as before to determine if the target is catastrophically killed. If it is killed, the incapacitation is scheduled again in TIME seconds.

Category 3 target elements have the following probalities calculated:

P.MKILL= (PA-PD)/PA

P.FKILL=PD*(1-TF)/PA

P.MFKILL=TF*PD*(1-TF)/PA

P.KKILL=TO*TF*PD/PA

Routine ATRIT is called to determine the appropriate level of incapacitation and to schedule it in the appropriate time.

The final set that the casualty might fall into is category 4. Here the element is assumed to be at least a mobility kill. The probability that the level of incapacitation is limited to this is:

P.ONLY=(PA-PD)/(PA-PD*(1-TF))

If the element is determined to have suffered a more severe incapacitation, the following probabilities are calculated:

P.FKILL=TF*PD*(1-TO)/(PA-PD*(1-TF))

P.KKILL=TO*TF*PD/(PA-PD*(1-TF))

Routine ATRIT is again called to determine appropriate levels of incapacitation and to schedule the occurence of

the event. Elements which are KKILLED are removed from the battle immediately. Elements which are MFKILLED are allowed to stay in the conflict until TIME expires, at which time event FINAL.DEATH is called to remove them from the battle. Other incapacitated elements are allowed to remain in the battle at their reduced levels of effectiveness.

Derivation and justification of the formulas and assumptions for casualty assessment are contained in the ASARS Battle Model documentation referenced in the bibliography.

5. Movement

The movement of infantry entities in the dismounted model uses the same methodology as the STAR Model. Movement of infantry forces is always one of the following types:

a. MOVEMENT MOUNTED/DISMOUNTED ON PRESELECTED ROUTES

b. ATTACKING INFANTRY FORCES DISMOUNTING

c. DISMOUNT/REMOUNT OF INFANTRY FORCES IN DEFENSIVE POSITIONS

A discussion of each type follows.

MOVEMENT MOUNTED/DISMOUNTED ON PRESELECTED ROUTES: The movement of infantry forces is closely coordinated with the movement of vehicles. Vehicle movement in this case is usually described in terms of moving from a position in the current platoon area to a position in the new platoon area.

The user must input the areas that each platoon is allowed to occupy during the battle. This data is in the array MOVE.DATA. Array MOVE.DATA is a 2-dimensional array, which is dimensioned as (THE NUMBER OF PLATOONS) by (3*(THE NUMBER OF ROUTES USED BY THE PLATOON)). A typical row of this array will look like this.

1 1 101 201 1

Column 1 specifies that this data is for the 1st platoon.

Column 2 specifies the platoon has one route to move on.

Column 3 specifies the first area it will occupy is AREA 101.

Column 4 specifies the next area the platoon will occupy is AREA 201.

Column 5 specifies that movement from AREA 101 to AREA 201 will be along route 1.

The above data is used to represent the situation depicted in Figure B-10 In Figure B-10 movement from AREA 101 to AREA 201 is represented by the straight line labeled ROUTE 1. Movement in some cases may be represented as a straight line. However, in most instances some change in the direction of the route is desired. Varying the direction of

DIRECT ROUTE BETWEEN POSITIONS

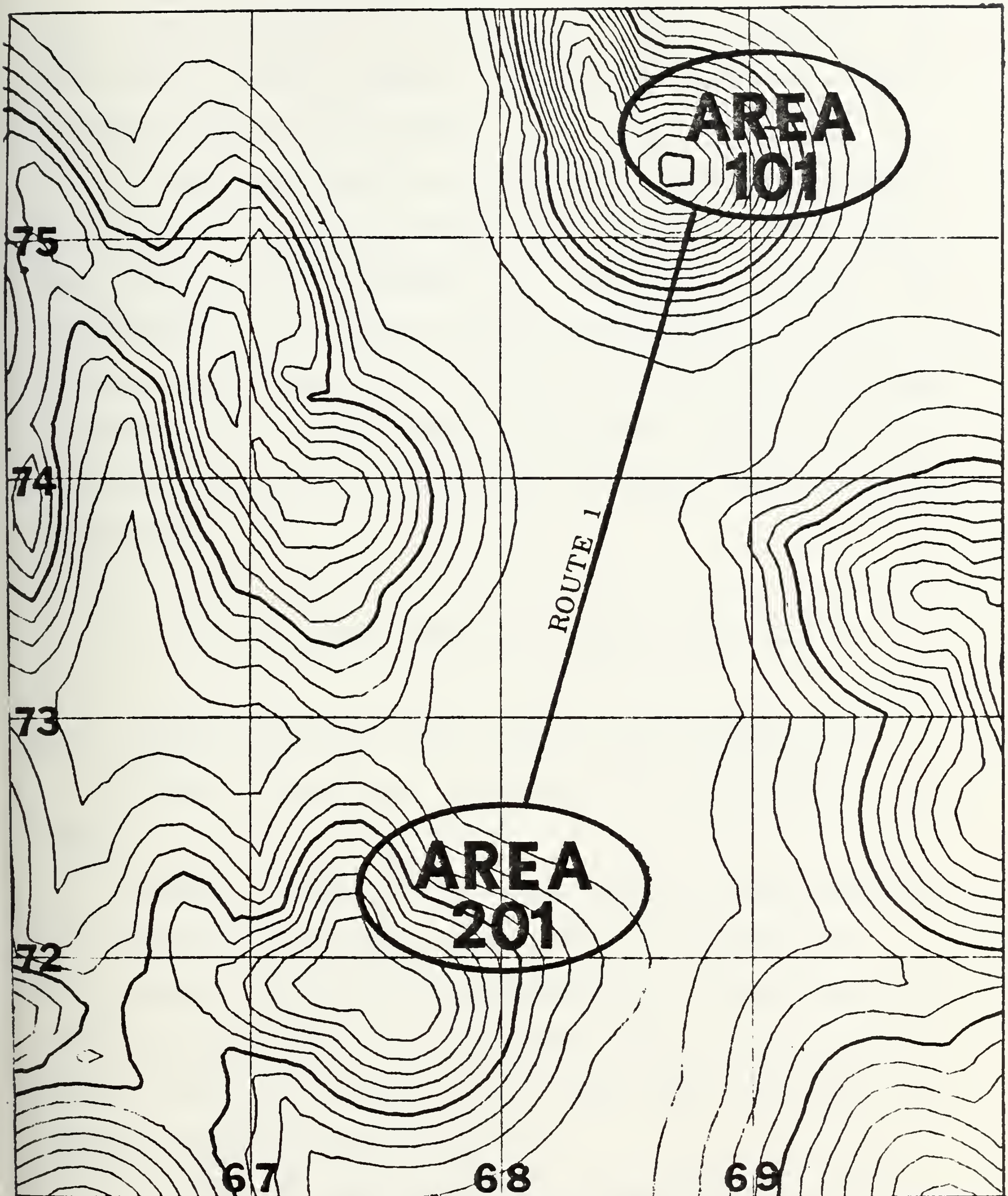


Figure B-10

the route requires an additional control measure. This additional control measure in STAR is referred to as a Movement Control Point (MCP).

A route may contain as many MCPs as necessary to obtain the desired movement pattern. Data containing the MCPs of each route must be entered as input to the model. This information is located in array ROUTE.DATA. This is a 2-dimensional array, the dimensions of which are (THE NUMBER OF ROUTES) by (3*(THE NUMBER OF MOVEMENT CONTROL POINTS)). The following is an example of a typical row from array ROUTE.DATA.

5 2 53505 98000 0 51500 97500 1

Column 1 specifies that this is route number one.

Column 2 specifies that there are two MCPs on this route.

Columns 3-5 specify the X-coordinate, Y-coordinate, and the formation requirement while moving to the new MCP.

Columns 6-8 specify the X-coordinate, Y-coordinate, and formation requirement for the second MCP on this route. Figure B-11 is an example of a route that is made up of several Movement Control Points.

As discussed above there is a requirement for platoons to travel in certain formations between each MCP. The

ROUTE WITH CHANGE OF DIRECTION BETWEEN POSITIONS

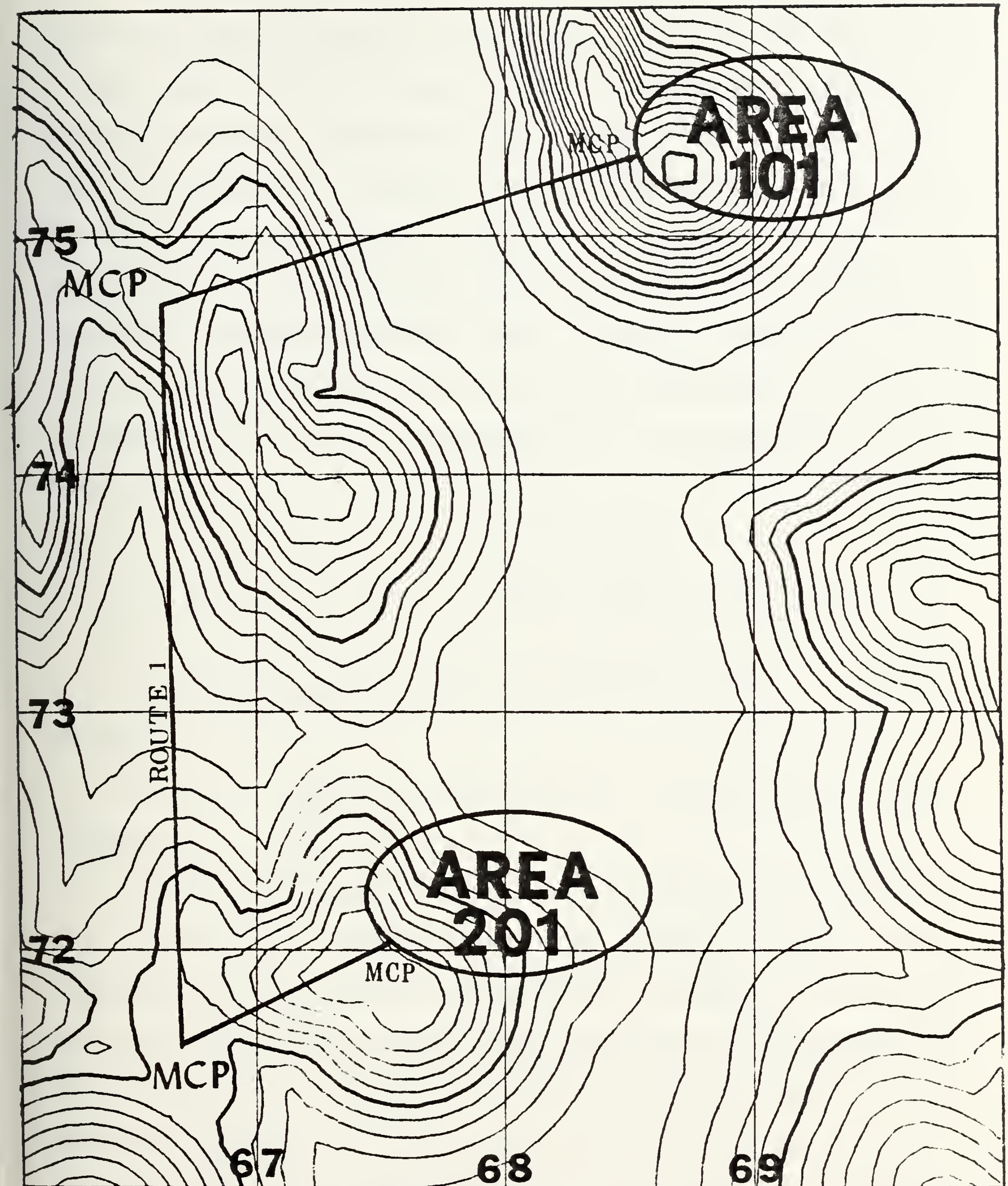


Figure B-11

formation that a platoon uses between each MCP can be varied on any route. Once a formation is dictated, the information on each formation specified must be entered as input data. This information is found in array FORM.OFFSET. This array is also a 2-dimensional array, the dimension of which is (THE NUMBER OF SPECIFIED FORMATIONS) by (THE NUMBER OF POSITIONS IN THE FORMATION). The following example is a row from array FORM.OFFSET which lists the formation offsets for a platoon of vehicles traveling in a line formation on a route.

1 5 50 0 0 0 -50 0 -100 0 100 0

Column 1 specifies this is formation number 1 for the program.

Column 2 specifies there are 5 positions in this formation.

Columns 3-4 specify the X and Y offsets for the vehicle with a FORM.POS (Formation Position) equal to one. Each succeeding pair of numbers specify the formation offsets for vehicles with formation positions which are in increments of one. A vehicle's formation position is the same as his POS.IN.PLT.AREA.

This completes the description of the movement of mounted infantry soldiers in the model. Infantry soldiers riding on

the vehicles are listed on the vehicle RIDER array. Additionally, a list of the members that are riding on the vehicle can be obtained by searching the set SQD.UNIT. Those entities with an ALIVE.DEAD state of two are riding on the squad vehicle. By accessing one of these two lists, gaming of individual members of a vehicle can be carried out. An example of this is the assessment of personnel casualties on a vehicle that has just been destroyed.

DISMOUNT/REMOUNT OF INFANTRY FORCES IN DEFENSIVE POSITIONS:

The dismounting of infantry soldiers in a defensive position takes place when a vehicle arrives at a new vehicle position and has come to a stop. In Figure B-12 a vehicle arriving at MPC 4 checks array POSITION to locate the vehicle position to which it is assigned in the platoon. The vehicle then moves on to its assigned vehicle position. If there are any riders on the vehicle, each rider will access array INF.POSITION to determine to which position in the platoon area he is to move. Movement from the vehicle to each position is in a straight line. In Figure B-12 the vehicle arriving at MPC 4 has a platoon position number of two and is carrying five riders. The vehicle moves to platoon POSITION 2. Each rider determines the coordinates of the position that corresponds to his position in the platoon. The five riders on this vehicle are assigned position

DISMOUNT AT A SUBSEQUENT POSITION

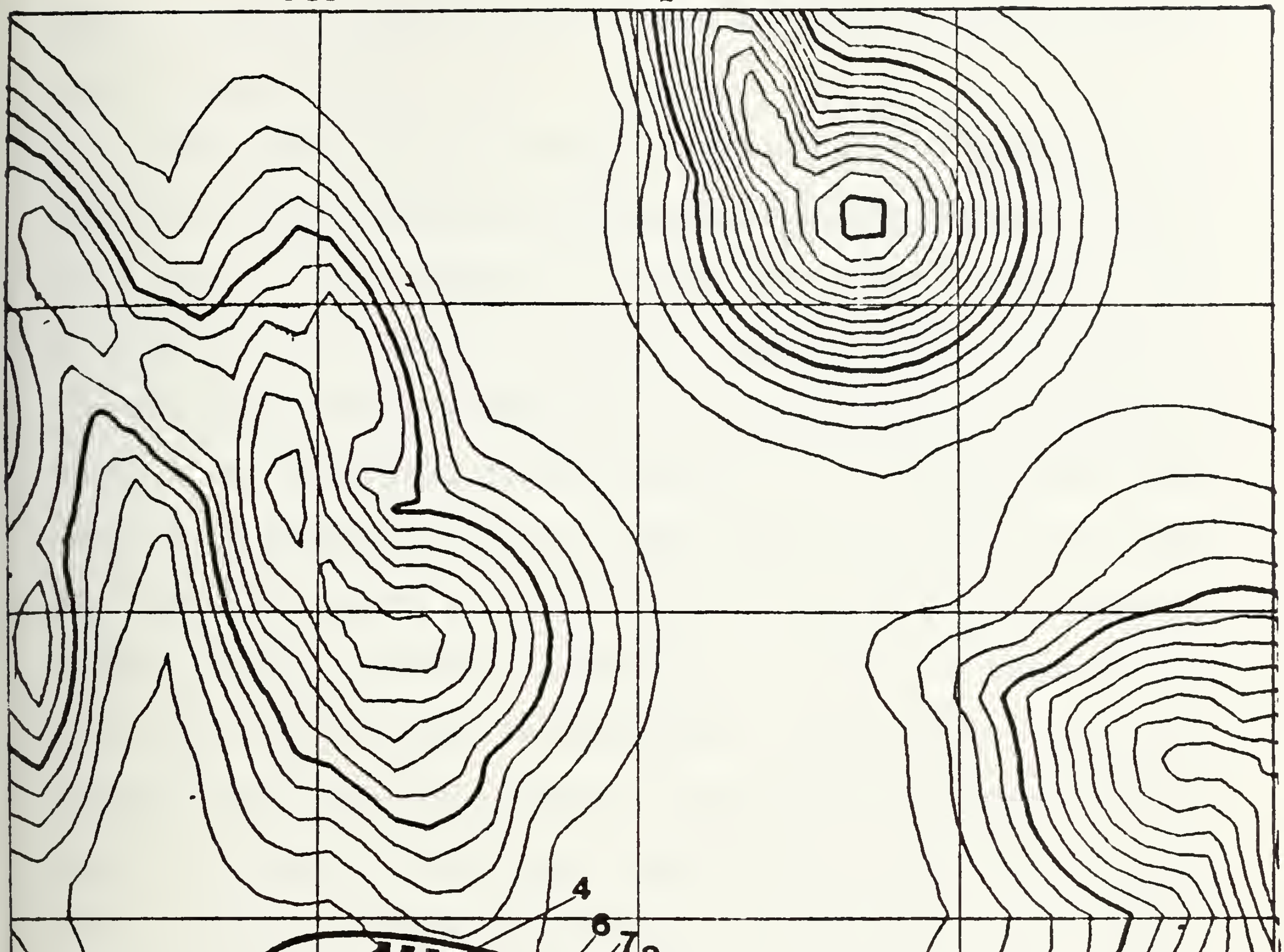


Figure B-12

numbers 4,6,7,8, and 10, and move directly from the vehicle to those positions.

The remounting of infantry soldiers is essentially the reverse of the dismounting procedure described above. When a unit has a requirement to move to a new location all dismounted entities in the unit search for a vehicle to remount. The first vehicle to which a soldier attempts to move is his assigned squad vehicle. If the assigned squad vehicle is capable of moving (not mobility killed), assigned squad members begin to move towards that vehicle to remount. If the vehicle suffers a mobility kill or a catastrophic kill while the squad is moving to the vehicle, the squad members stop in place, look for the vehicle in their platoon that is closest to them and begin moving to that vehicle. Vehicles wait a specified amount of time for the infantry (determined by the user). Any soldiers that do not arrive at the vehicle in time are left behind and attempt to move to their next defensive position on foot. Movement on foot is along the same route the vehicles use.

THE DISMOUNT OF ATTACKING FORCES: In the current version of the dismounted model, the dismounting of attacking forces is controlled in routine REDACT. This routine models a variety of possible tactical options an attacking unit has available for use during the attack. An example of these options is

to go into a hasty defense. The action that takes place is a function of the attrition rate of the attacking force. As an example the user can direct an attacking force to go into a hasty defense when fifty percent of the battalion has been attrited. It is important to note that future versions of the model will not use attrition rate as the only criteria to dismount attacking forces. Future versions will dismount forces as a function of the following:

a. Force ratio

b. Attrition rate

c. Distance to the objective

d. Predesignated dismount check points

When it is determined that it is time to dismount and continue the attack all vehicles carrying infantrymen stop to let them dismount. Each infantry squad moves to the front of the vehicle then moves into a line formation for the remainder of the attack. In designing the dismounted attack the user may designate which vehicle types and which units will go forward with the infantry. Vehicles that do not move with the dismounted force support by fire. The direction that a dismounted squad moves is the same as the direction that its vehicle was moving at the time of the dismount.

One final topic to be covered under the subject of movement is the methodology used to control the speed of

each entity. The speed capabilities of entities are required input parameters to the STAR model. Two global arrays contain the speed limits of each system type and weapon type in the simulation.

The array LIM.SPDS is a 2-dimensional array, the dimension of which is (THE NUMBER OF SYSTEMS) by (13). This array is accessed when an entity is involved in a mounted attack. A typical row of array LIM.SPDS follows.

2 8 .09 -.09 4 5 6 .2 .54 .76 -4. -4. -4.

Columns 1-2 specify the system type and weapon type of the entity. This entity is a BMP.

Column 3 specifies that if the slope of the terrain that the vehicle is moving over is greater than .09 it is considered moving 'UPSLOPE'.

Column 4 specifies that if the slope of the terrain that the vehicle is moving over is less than -.09 it is considered moving 'DOWN SLOPE'. If neither case applies, the vehicle is considered moving on a 'LEVEL SLOPE'.

Columns 5-7 specify the maximum speed limits this vehicle can attain moving up slope, level slope, and down slope respectively.

Columns 8-10 specify the acceleration limits for the types of slopes.

Column 11-13 specify the deceleration limits of the vehicle for the three types of slopes.

The second array used to control the speed of entities is the array DSMTD.SPDS. This array is dimensioned the same as LIM.SPDS. The only difference is that this array is used when an entity is part of a dismounted attacking force. This array will slow vehicle movement down to a rate compatible with the infantry. The following is an example of the speed limits of a BMP in the dismounted attack. Columns are used to specify the same conditions as array LIM.SPDS.

2 8 .09 -.09 3 3 3 .2 .54 .76 -.3 -.3 -.3

APPENDIX C. MODIFIED ROUTINES AND EVENTS FROM BASIC STAR

1. PREAMBLE

Purpose

The preamble provides the compiler with definitions regarding entities, attributes, and sets; events and routines; background mode, type, and dimensionality; global variables and arrays.

Modifications

The following additions have been made to the preamble. For additional information refer to the thesis section listed below.

Events (Appendix D)

EVENT MAX.WAIT.TIME

Routines (Appendix D)

CALL.TO.REMOUNT	DISMOUNT
INF.ARRIVAL	INF.DEST
REMOUNT	TANK.OVERWATCH
DISMTD.CARDIO	BRST.FIRE
INCAPACITATE	INF.COMPUTE
RES1	

Sets (Appendix F)

SQD.UNIT

Attributes (Appendix F)

VEH.TACTIC	GRNDATK

SQDVEH	SCH.TO.MOVE
R.D.STATUS	N.AB
N.ARM	N.HD
N.PLE	N.TH
P.AB.INCAP	P.ARM.INCAP
P.BODY.INCAP	P.HD.INCAP
P.PLE.INCAP	P.TH.INCAP

Global Variables (Appendix F)

KILLED.MOUNTED	AB.HT
MAXCARR	INF.CAS.ASSMT
AB.TH.WIDTH	ABDO.HT
THORAX.HT	PLE.HT
PLE.WIDTH	PCT.INCAP
A.CSLTY.DATA	AB.IN
ARM.IN	B.CSLTY.DATA
C.CSLTY.DATA	H.IN
PLE.IN	TH.IN

Entities (Appendix F)

SQUAD.LEADER

2. <u>Routine RED.CREATE</u>

Purpose

Routine RED.CREATE is used to create temporary TANK entities at designated times throughout the execution of the program. The routine is called during the execution of event NEW.FORCES. This indicates the creation of new attacking

forces.

Modifications

The following eleven attributes must be assigned values from the input data:

NAME(TANK)	CO(TANK)
SYS.TYPE(TANK)	BN(TANK)
WPN.TYPE(TANK)	SQDVEH(TANK)
SEC(TANK)	PLTLDR(TANK)
PLT(TANK)	COCDR(TANK)
ALIVE.DEAD(TANK)	

Each red entity is additionally filed in the assigned set SQD.UNIT. When creating the members of a squad, the squad vehicle must be created before any other members of the squad are created. This allows access to the vehicle rider list when a squad member is created mounted in the execution of the program.

3. Routine BL.CREATE

Purpose

Routine BL.CREATE is used to create temporary TANK entities at designated times throughout the execution of the program. The routine is called during the execution of event BLU.FORCES.

Modifications

The following twelve attributes must be assigned initial values from the input data:

NAME(TANK)	BN(TANK)
SYS.TYPE(TANK)	SQDVEH(TANK)
WPN.TYPE(TANK)	PLTLDR(TANK)
SEC(TANK)	COCDR(TANK)
PLT(TANK)	AREA.START(TANK)
CO(TANK)	ALIVE.DEAD(TANK)

Each entity is additionally filed in the assigned set SQD.UNIT. When creating the members of any squad it is necessary that the first entity created in each squad is the squad vehicle. This is required because some entities are created with the attribute ALIVE.DEAD=2, which indicates that the entity is mounted. This requires the entity to search for it's assigned SQDVEH(TANK) and dynamically increases the size of the vehicle rider list.

4. Routine INIT.POS

Purpose

Routine INIT.POS is used to assign each TANK entity an initial X-coordinate, Y-coordinate, and a direction of movement on the terrain. Additionally, routine BEST.POS is called to assign each entity a position number within the platoon. The entity's position in a platoon area dictates exactly which position in any platoon area the entity will occupy.

Modifications

Separate arrays are reserved which contain vehicle

positions and dismounted infantry positions. Therefore a check is made in this routine to determine if the entity is a vehicle or an infantryman. The proper position array is then accessed.

5. <u>Routine BEST.POS</u>

Purpose

Routine BEST.POS is called from routine INIT.POS to assign each TANK entity a position number in the platoon that the entity is a member. This position number is then used to access position information in routine INIT.POS.

Modifications

Because the infantry version of STAR may be executed with both vehicles and dismounted elements, it is necessary to maintain two separate sets of platoon position numbers. Therefore a check is made to determine if the entity is a vehicle or an infantryman. The proper position number is then assigned to the entity.

6. <u>Routine BN.GO</u>

Description

This routine is called from routine ACTION when an event in the simulation has taken place which calls for the movement of a blue battalion to a new defensive position. One or more of the following parameters are used to trigger a movement.

(1) Attrition rate.

(2) Force ratio.

(3) Range to the enemy.

Modification

This routine causes all dismounted elements to select a vehicle to remount. A battalion movement time will then be scheduled for each vehicle in the battalion. This allows all dismounted elements a fixed amount of time to select and to remount a vehicle. The amount of time a vehicle waits is a user input parameter of the model called MAX.REMOUNT.WAIT.TIME. The following routines are also designed in the same manner as routine BN.GO. The only difference in each routine is the size of the unit that is moving to a new location.

Routine SEC.GO (move a section)

Routine VEH.GO (move a vehicle)

Routine PLT.GO (move a platoon)

Routine CO.GO (move a company)

Routine OTHER.GO (move all unmonitored systems)

7. Routine MOVE.LIMITS

Description

Routine MOVE.LIMITS is called from routine MOVE to determine a vehicle's maximum speed, acceleration rate, and deceleration rate. These parameters are a function of the weapon type, the system type, and the slope of the terrain. Slope is defined as either up, down, or level. The amount of

slope necessary to qualify a slope as up or down is a user input parameter.

Modifications

The routine now returns speed limits that are also a function of the tactical situation as well as the slope of the terrain. If a vehicle is moving with a dismounted infantry element the vehicle speed limits are obtained from the array DSMTD.SPDS. This array contains speed limits which coordinate the movement of vehicles and dismounted infantry elements. In the other case where vehicles are moving in a mounted attack, the speed limits are obtained from the array LIM.SPDS. The temporary attribute GRND.ATK is used to identify the state of the attack.

GRND.ATK=1 (vehicle is part of a dismounted attack)

GRND.ATK=0 (vehicle is moving in a mounted attack)

8. Routine MOVE

Description

Routine MOVE is called from routine LOC to update the location of any entity to the current time of the simulation.

Modifications

The movement routine is now used to move an infantry soldier from a point A in a straight line to a point B. This is accomplished through the use of two temporary attributes MV.STATE and ROUTE. Whenever an infantry soldier desires to

move to a new location the value of MV.STATE is set to two. This prevents the movement routine from calling the route selection routine. As a result, the attribute ROUTE of the entity is always zero. This situation causes the movement routine to call routine INF.DEST. Routine INF.DEST returns a set of coordinates to which the soldier is to move. If the soldier is remounting a vehicle, the coordinates of a vehicle in the platoon are returned. If there are no vehicles available in the platoon to be remounted, the coordinates X=0, Y=0 are returned. This causes the movement routine to call the route selection routine to move the soldier along a preplanned route to his next defensive position. When a soldier is dismounting a vehicle, the coordinates of an infantry fighting position are returned to the movement routine.

APPENDIX D. NEW ROUTINES AND EVENTS CREATED IN STAR

1. Routine INF.DEST

Description

Routine INF.DEST is called from routine MOVE when one of the following two situations occurs:

a. When a vehicle has reached a new defensive position this routine returns to the movement routine the preselected coordinates that each soldier is to occupy.

b. When a unit is going to leave a defensive position to move to a new location this routine returns to routine MOVE the coordinates of the vehicle that the soldier is to remount. Each soldier first checks to see if his assigned squad vehicle is operative. If the soldier's vehicle is inoperative, a check of the other vehicles in the platoon is made to determine if there are other vehicles the soldier may remount. If there are vehicles available to remount, the soldier selects that vehicle which is closest to his current position. If there are no vehicles available in the platoon for dismounted soldiers to remount, those soldiers move along the platoon route to the next defensive position.

Local Variables

B - An integer variable used to save the pointer of

a tank.

X.DEST, Y.DEST - Real variables representing the coordinates that a dismounted soldier is to move towards.

A - The pointer to the infantry entity that is looking for a position.

DIST.TO.A - Real variable representing the distance between a vehicle and an infantryman.

SUBTANK - Local variable representing the pointer to an entity TANK.

Coding and Brief Explanation

```
ROUTINE INF.DEST(A) YIELDING X.DEST,Y.DEST
DEFINE A,B,I,J,K AS INTEGER VARIABLES
DEFINE SUBTANK AS AN INTEGER VARIABLE
DEFINE X.DEST,Y.DEST AS REAL VARIABLES
DEFINE DIST.TO.A,MINDIST AS REAL VARIABLES
LET B=0
LET MINDIST=2000.0
IF R.D.STATUS(A)=1 GO TO DISMOUNT ELSE
''IF A SOLDIER IS REMOUNTING HIS OWN VEHICLE DO THIS
IF SYS.TYPE(A) NE 3 RETURN
''IF YOUR VEHICLE IS INOPERATIVE SEARCH YOUR PLATOON
IF (ALIVE.DEAD(SQDVEH(A))=0 AND MKILL(SQDVEH(A)) LT 1
AND MFKILL(SQDVEH(A)) LT 1)
LET X.DEST=X.CURRENT(SQDVEH(A))
LET Y.DEST=Y.CURRENT(SQDVEH(A))
```

```
RETURN
ELSE
''HERE HE IS GOING TO CHOOSE AN ALTERNATE VEHICLE
''FROM THE VEHICLES IN HIS PLATOON
FOR EACH SUBTANK IN PLT.UNIT(PLT(A)) WITH
SYS.TYPE(SUBTANK)=2
AND WPN.TYPE(SUBTANK)=3 OR WPN.TYPE(SUBTANK)=8) AND
ALIVE.DEAD(SUBTANK) NE 1, DO
LET DIST.TO.A=SQRT.F((X.CURRENT(A)-X.CURRENT(SUBTANK)
AND (Y.CURRENT(A)-Y.CURRENT(SUBTANK)**2)
IF DIST.TO.A LT MINDIST
LET MINDIST=DIST.TO.A
LET X.DEST=X.CURRENT(SUBTANK)
LET Y.DEST=Y.CURRENT(SUBTANK)
LET B=SUBTANK
LET SEC(A)=SEC(B)
LET SQDVEH(A)=SQDVEH(B)
REMOVE A FROM SQD.UNIT(SEC(A))
FILE A IN SQD.UNIT(SEC(B))
ALWAYS
LOOP
IF MINDIST=2000.0
LET MV.STATE(A)=1
LET T.SPD(A)=TIME.V
LET AREA.START(A)=AREA.START(SQDVEH(A))
LET AREA.END(A)=AREA.END(SQDVEH(A))
```

```
40  LET X.DEST=0
41  LET Y.DEST=0
42  ALWAYS
43  RETURN
44  'DISMOUNT'
45  LET I=PLT(A)
46  LET K=POS.IN.PLT.AREA(A)*3
47  FOR J=1 TO DIM.F(INFANTRY.POSITION(I,*,*)) WITH
48  INFANTRY.POSITION(I,J,1) EQUALS AREA.END(A),DO
49  LET X.DEST=INFANTRY.POSITION(I,J,K-1)
50  LET Y.DEST=INFANTRY.POSITION(I,J,K)
51  LOOP
52  RETURN
53  END
```

Lines 1-5 define the routine and the local variables.

Lines 6-7 initialize local variables.

Lines 8-9 check to determine if this is a remount or a dismount.

Lines 10-12 determine if the element's assigned squad vehicle is still operational.

Lines 13-15 return the coordinates of the squad vehicle.

Lines 17-34 check all other vehicles in a given platoon to determine if any are available to carry infantrymen. If there are vehicles available the coordinates of the vehicle closest to the soldier is returned as the desired destination.

Lines 35-43 return a destination of X.DEST=0 and Y.DEST=0 when there are no vehicles available to remount. This causes the soldier to move to his next fighting position in the movement routine.

Lines 44-51 determine the coordinates of the infantry position that a dismounting soldier is to occupy at a new defensive position.

Lines 52-53 return control to the system timer.

2. <u>Routine CALL.TO.REMOUNT</u>

Description

Routine CALL.TO.REMOUNT is called from routines which direct units of varying size to move to new positions. If the moving unit contains dismounted elements this routine is called to assign to attributes the proper values necessary to allow the infantry to move from a defensive position to a selected squad vehicle.

Local Variables

A - The pointer to the soldier that needs to move from an infantry position to a vehicle.

Coding and Brief Explanation

```
1 ROUTINE CALL.TO.REMOUNT(A)
2 DEFINE A AS AN INTEGER VARIABLE
3 LET ROUTE(A)=0
4 LET MV.STATE(A)=2
```

```
5  LET T.SPD(A)=TIME.V
6  LET R.D.STATUS(A)=0
7  RETURN
8  END
```

Lines 1-2 define the routine and the local variables.

Lines 3-6 update attributes to allow this entity to move on a straight line from his present position to a new location. R.D.STATUS(A)=0 indicates that this is a move designed to remount the soldier. The setting of ROUTE(A)=0 eliminates the requirement that this entity move along a predetermined route in the movement routine.

Lines 7-8 return control to the system timer.

3. <u>Routine DISMOUNT</u>

Description

Routine DISMOUNT dismounts red and blue infantry entities from their squad vehicles. This event is called by events HASDEF, DEFEND, ACTION, AND REDACT and routine TALLY.HIT.STATE.

Local Variables

A - The pointer to the vehicle that is to dismount riders.

WHO - An integer variable indicating which event or routine called for this dismount. WHO IS Assigned the following values:

HASDEF - 1

DEFEND - 2

TALLY.HIT.STATE - 3

REDACT - 4

ACTION - 4

STOP.OR.GO - An integer variable which indicates whether a vehicle is to stop or to begin to move again.

STOP.OR.GO=1 (Stop)

STOP.OR.GO=2 (Go)

SUBTANK - Local variable representing the entity TANK.

Coding and Brief Explanation

```
ROUTINE DISMOUNT(A,WHO)
DEFINE WHO,SUBTANK,STOP.OR.GO,J,JJ AS INTEGER VARIABLES
IF CARR(A)=0 RETURN ELSE
IF RIDER(NAME(A),1)=0 GO TO OUT ELSE
SUBSTITUTE THESE 9 LINES FOR ATTRIBUTES
LET ALIVE.DEAD(SUBTANK)=0
LET T.SPD(SUBTANK)=TIME.V
LET DIR.OF.MVMT(SUBTANK)=DIR.OF.MVMT(A)
LET PRI.DIR(SUBTANK)=PRI.DIR(A)
LET LST.DIR(SUBTANK)=LST.DIR(A)
LET AREA.START(SUBTANK)=AREA.START(A)
LET AREA.END(SUBTANK)=AREA.END(A)
LET DIR.ON.RT(SUBTANK)=DIR.ON.RT(A)
LET CARR(A)=CARR(A)-1
IF COLOR(A) EQ BLUE GO TO BLUE.DISMOUNT.TROOPS ELSE
```

```
''DISMOUNT RED FORCES
CALL TANK.OVERWATCH(A)
LET STOP.OR.GO=1  ''CAUSE THE VEHICLE TO STOP
CALL STOP.DISMOUNT(A,STOP.OR.GO)
''CREATE THE PROPER RED FORMATION ORIENTED TOWARDS THE
OBJECTIVE
LET J=0
FOR EACH SUBTANK IN SQD.UNIT(SEC(A)) WITH
ALIVE.DEAD(SUBTANK)=2, DO
LET J=J+1
LET X.CURRENT(SUBTANK)=X.CURRENT(A)+(OFFSET(J)*
SIN.F(DIR.OF.MVMT(A)))
LET Y.CURRENT(SUBTANK)=Y.CURRENT(A)+(OFFSET(J)*
COS.F(DIR.OR.MVMT(A)))
GO TO ONE,TWO,THREE,FOUR PER WHO
'ONE'
LET MV.STATE(SUBTANK)=5  ''STOPPED IN HASTY DEFENSE
LET DEFNUM(SUBTANK)=2  ''LOWER BODY IN DEFILADE
CALL LOC(SUBTANK)
CALL HIDER(SUBTANK)
GO TO FIX.ATTRIBUTES
'TWO'
''RED DEFEND CURRENTLY NOT USED
'THREE'  ''CALLED FROM TALLY.HIT.STATE
LET MV.STATE(SUBTANK)=3  ''STOPPED ON ROUTE
LET DEFNUM(SUBTANK)=2  ''LOWER BODY IN DEFILADE
```

```
GO TO FIX.ATTRIBUTES
'FOUR'  ''REDACT SAYS DISMOUNT AND FIGHT
LET MV.STATE(SUBTANK)=2  ''MOVING DISMOUNTED
LET DEFNUM(SUBTANK)=5
CALL LOC(SUBTANK)
ATTRIBUTES
LET JJ=RIDER(NAME(A),J)
IF JJ=0 CYCLE ELSE
IF CARR(A)=0
RELEASE RIDER(NAME(A),*)
RESERVE RIDER(NAME(A),*) AS 1
GO TO OUT ELSE
LOOP
GO TO OUT
'BLUE.DISMOUNT.TROOPS'  ''DISMOUNT BLUE FORCES
CALL TANK.OVERWATCH(A)
LET STOP.OR.GO=1  ''CAUSE VEHICLE TO STOP
CALL STOP.DISMOUNT(A,STOP.OR.GO)
GO TO HASTY,DEFEND,TALLY,OUT PER WHO
'HASTY'
FOR EACH SUBTANK IN SQD.UNIT(SEC(A)) WITH
ALIVE.DEAD(SUBTANK)=2, DO
LET J=J+1
LET X.CURRENT(SUBTANK)=X.CURRENT(A)+(OFFSET(J)*
SIN.F(DIR.OF.MVMT(A)))
LET Y.CURRENT(SUBTANK)=Y.CURRENT(A)+(OFFSET(J)*
```

```
COS.F(DIR.OF.MVMT(A)))
LET CARR(SUBTANK)=0
LET MV.STATE(SUBTANK)=5
LET DEFNUM(SUBTANK)=2
CALL LOC(SUBTANK)
CALL HIDER(SUBTANK)
ATTRIBUTES
LOOP
RELEASE RIDER(NAME(A),*) RESERVE
RIDER(NAME(A),*) AS 1
GO TO OUT
'DEFEND'  ''CALL FROM DEFEND
LET J=1
FOR EACH SUBTANK IN SQD.UNIT(SEC(A)) WITH
ALIVE.DEAD(SUBTANK)=2, DO
LET JJ=RIDER(NAME(A),J)
LET RIDER(NAME(A),J)=00
IF JJ=0 CYCLE ELSE
IF CARR(A)=0 GO TO OUT ELSE
LET MV.STATE(SUBTANK)=2
LET DEFNUM(SUBTANK)=5
CALL HIDER(SUBTANK)
LET ROUTE(SUBTANK)=0
LET NEXT.MCP(SUBTANK)=0
LET R.D.STATUS(SUBTANK)=1
LET X.CURRENT(SUBTANK)=X.CURRENT(A)
```

```
LET Y.CURRENT(SUBTANK)=Y.CURRENT(A)
LET Z.CURRENT(SUBTANK)=Z.CURRENT(A)
LET J=J+1
ATTRIBUTES
LOOP
RELEASE RIDER(NAME(A),*) RESERVE
RIDER(NAME(A),*) AS 1
GO TO OUT
'TALLY'  ''CALLED BY TALLY.HIT.STATE
LET J=1
FOR EACH SUBTANK IN SQD.UNIT(SEC(A)) WITH
ALIVE.DEAD(SUBTANK)=2, DO
LET JJ=RIDER(NAME(A),J) LET RIDER(NAME(A),J)=0
IF JJ=0 CYCLE ELSE
IF CARR(A)=0 GO TO OUT ELSE
LET X.CURRENT(SUBTANK)=X.CURRENT(A)+(OFFSET(J)*
SIN.F(DIR.OF.MVMT(A)))
LET Y.CURRENT(SUBTANK)=Y.CURRENT(A)+(OFFSET(J)*
COS.F(DIR.OF.MVMT(A)))
'FIX.ATTRIBUTES'
ATTRIBUTES LOOP
RELEASE RIDER(NAME(A),*) RESERVE
RIDER(NAME(A),*) AS 1
'OUT'
RETURN
END
```

Lines 1-2 define the routine and the local variables.

Lines 3-4 check to insure that there is a rider to dismount.

Lines 5-14 define a block of coding that will be used repetitively in this routine and is to be substituted in place of the key word ATTRIBUTES when it appears in the routine.

Line 15 checks to see if this is a red or a blue dismount.

Lines 16-53 are used to dismount red infantrymen from individual squad vehicles. This section of code is further broken down into four subsets which will set each squad member's attributes to the proper values required by the event causing this dismount. Red dismounts occur whenever any of the following events take place in the simulation:

a. Red attacker is forced to dismount into a hasty defense.

b. A red attacking vehicle suffers a mobility kill or a mobility and firepower kill.

c. Red is forced to dismount and to defend his vehicle.

d. Red forces have chosen to dismount to continue the attack.

Lines 54-105 are used to dismount the Blue forces from thier squad vehicles. This section is also subdivided into smaller sections designed to set attributes of each soldier to those values required by the situation calling for the dismount. There are currently three situations which cause a blue

vehicle to dismount soldiers.

a. Blue vehicles arrive at a new defensive position.

b. Blue vehicle suffers a mobility kill or a mobility and firepower kill.

c. Blue is forced to go into a hasty defense between positions.

Lines 106-107 return control to the system timer.

4. <u>Routine INF.ARRIVAL</u>

Description

Routine INF.ARRIVAL is called from routine MOVE if during some phase of a move a soldier arrives as his vehicle or his infantry fighting position. The value of the entity's temporary attribute R.D.STATUS is used to distinguish between the two situations. R.D.STATUS equal to zero indicates that the entity has arrived at a vehicle and is ready to remount. Otherwise R.D.STATUS is equal to one which means the entity has arrived at an infantry fighting position.

Local Variables

A - The pointer to the infantryman arriving at a selected location.

I,K,J - Integer variables used to access the proper values from the three dimensional array used to store locations of infantry positions.

GRAD.X, GRAD.Y - Real variables representing the

gradient in the X and Y directions at the current location of an infantryman used in routine MOVE to determine the speed of a soldier.

Coding and Brief Explanation

```
ROUTINE INF.ARRIVAL(A)
DEFINE GRAD.X,GRAD.Y AS REAL VARIABLES
DEFINE I,J,K,A AS INTEGER VARIABLES
IF R.D.STATUS(A) EQ 0
CALL REMOUNT(A,2)
RETURN
ELSE
LET R.D.STATUS(A)=0
LET DEFNUM(A)=3
LET MV.STATE(A)=0
LET T.SPD(A)=TIME.V
LET I=PLT(A)
LET K=POS.IN.PLT.AREA(A)*3
FOR J=1 TO DIM.F(INFANTRY.POSITION(I,*,*))
WITH INFANTRY.POSITION(I,J,1) EQUALS
AREA.END(A), DO
LET DIR.OF.MVMT(A)=INFANTRY.POSITION(I,J,K+1)
LET PRI.DIR(A)=DIR.OF.MVMT(A)
LOOP
CALL ELEVG(X.CURRENT(A),Y.CURRENT(A)) YIELDING
Z.CURRENT(A),GRAD.X,GRAD.Y
RETURN
```

20 END

Lines 1-3 define the routine and the local variables.

Line 4 determines whether this is an infantryman arriving at a new position or his squad vehicle.

Lines 5-6 call routine REMOUNT if this is a soldier arriving at a vehicle.

Lines 7-11 set attributes of the soldier to values causing him to stop moving.

Lines 12-17 select the proper position for the infantryman to occupy in the platoon defensive position.

Line 18 updates coordinates of the infantryman.

Lines 19-20 return control to the system timer.

5. <u>Routine REMOUNT(A,WHO)</u>

Description

Routine REMOUNT is called from event MAX.WAIT.TIME and routine INF.ARRIVAL. The routine determines if there is room on a vehicle for a soldier to mount. If there is room on the vehicle the soldier's attributes are updated to reflect that he is riding on a vehicle. If there is no room to ride the soldier attempts to move to his next defensive position by foot using his platoon preplanned route.

Local Variables

A - Integer variable used for the pointer to the soldier remounting a vehicle.

WHO - Integer variable representing the routine that called the remount.

Coding and Brief Explanation

```
ROUTINE REMOUNT(A,WHO)
DEFINE A,WHO,I AS INTEGER VARIABLES
IF ALIVE.DEAD(A)=1 RETURN END
IF SYS.TYPE(A) NE 3 RETURN END
IF CARR(SQDVEH(A)) GT MAXCARR OR MKILL(SQDVEH(A))
GE 1 OR MFKILL(SQDVEH(A)) GE 1 OR ALIVE.DEAD(A) EQ 1
RETURN
ELSE
RESERVE TEMP.RIDER(*) AS DIM.F(RIDER(SQDVEH(A)),*))
FOR I=1 TO DIM.F(RIDER(NAME(SQDVEH(A)),*)),
LET TEMP.RIDER(I)=RIDER(NAME(SQDVEH(A)),I)
RELEASE RIDER(NAME(SQDVEH(A)),*)
RESERVE RIDER(NAME(SQDVEH(A)),*) AS CARR(SQDVEH(A))+1
FOR J=1 TO DIM.F(TEMP.RIDER(8)),
LET RIDER(NAME(SQDVEH(A)),J)=TEMP.RIDER(J)
LET RIDER(NAME(SQDVEH(A)),CARR(SQDVEH(A))+1)=A
RELEASE TEMP.RIDER(*)
LET CARR(SQDVEH(A))=CARR(SQDVEH(A)+1
LET X.CURRENT(A)=0.0
LET Y.CURRENT(A)=0.0
LET ALIVE.DEAD(A)=2
LET MV.STATE(A)=0
LET DEFNUM(A)=1
```

```
23  LET R.D.STATUS(A)=1
24  RETURN
25  END
```

Lines 1-2 define the routine and the local variables.

Lines 3-6 check to determine whether the vehicle is still operational and if there is space available for riders in the vehicle.

Lines 7-24 add the soldier to the vehicle list. The soldier's attributes are then updated to reflect that he is riding on a vehicle.

Line 25 returns control to the system timer.

6. Routine DISMTD.CARDIO

Description

This routine receives the pointer values for two elements, the range between the element A, and a random number X. Element B must be a dismounted element. This routine then returns a detection time for element A to detect element B. If the value of the detection time, DET.TIME, is greater than DELTA.T then this detection will not occur this STEP.TIME. In calculating the detection time routine DISMTD.CARDIO takes into account the following factors:

a. The probability that the observer was looking in the right direction.

b. The terrain complexity of the terrain in which

the entity to be detected lies.

c. The number of confusing background figures in the area where the entity to be detected lies.

d. The velocity of the element to be detected.

e. The crossing velocity of the element to be detected.

f. The percent fully exposed of the element to be detected.

g. The apparent range factor of the element to be detected.

Local Variables

ANGLE - A real variable indicating the difference between the direction that the observer is currently looking and the direction from the observer to the target.

AT - A real computational variable equal to 1/2 the value of pi which is used to calculate the probability that the observer was looking in the sector containing the target.

BT - A real computational variable equal to 3/8 the value of pi which is used to calculate the probability that the observer was looking in the sector containing the target.

CBAR - An attribute indicating the number of confusing background forms in the vicinity of the target element.

CROSSING.VEL.FACTOR - A real variable which represents the value of X.VELOCITY transformed to be used in computing the mean of a log-normal detection time distribution.

DD - A real dummy variable set equal to PCT.VIS.

MEAN.DET.TIME - A real variable indicating the mean detection time parameter for use with the log-normal distribution to derive the actual detection time.

MT - A real computational variable equal to six divided by pi which is used to calculate the probability that the observer was looking in the sector containing the target.

P.HAT - An integer variable used to hold the value of the target element's PI.HAT attribute.

P.SUB.K - A real variable indicating the probability that the observer is looking in the k-th sub section of a cardioid.

PER.FULL.EXPO - A real variable equivalent to the percent fully exposed which accounts for the number of confusing forms in the vicinity of the target.

PRI.DIR - A real variable which adjusts the range for the percent of the target which is fully exposed. The result is an apparent range factor.

RGE.FACTOR - A real variable which represents the equivalent range to a standard sized fully exposed target. The result is an apparent range factor.

TC.FACTOR - A real variable dependent upon the complexity of the terrain in which the element lies.

TGT.ELEMENT - A real variable indicating the angle from the observer to the element being detected. This is measured from East.

X.VELOCITY - A real variable representing the apparent velocity of the target as it crosses in front of the observer. This accounts for the target moving at an angle across the front of the observer.

ZL - A real variable indicating a zero limit.

Coding and Brief Explanation

```
1  ROUTINE DISMTD.CARDIO(A,B,R,PCT.VIS,X) YIELDING DET.TIME
2  DEFINE R,PCT.VIS AS REAL VARIABLES
3  DEFINE A AND B AS INTEGER VARIABLES
4  DEFINE DD,ZL,TGT.ELEMENT,P.SUB.K,ANGLE,MT,BT,AT,X,
   X.VELOCITY,TC.FACTOR,PER.FULL.EXPO,RGE.FACTOR,
   CROSSING.VEL.FACTOR,MEAN.DET.TIME,DET.TIME AS
   REAL VARIABLES
5  DEFINE P.HAT AS AN INTEGER VARIABLE
6  IF R LE 10. LET DET.TIME=1.0
7  GO TO OUT
8  ELSE
9  IF R GT 1200. LET DET.TIME=99
10 GO TO OUT
11 ELSE
12 LET ZL=0.000001
```

```
LET DD=PCT.VIS
IF DD LE 0.0 LET DD=ZL
ALWAYS
LET TGT.ELEMENT=ARCTAN((Y.CURRENT(B)-Y.CURRENT(A),
(X.CURRENT(B)-X.CURRENT(A)))
IF AREA(A)=360.  ''HAVEN'T DETECTED ANY TARGETS IN
YOUR SECTOR
GO TO FULL.CARDIOID
ELSE
LET P.SUB.K=30./AREA(A)  ''REDUCE THE SEARCH AREA
GO TO CONTINUE
'FULL.CARDIOID'
LET ANGLE=ABS.F(TGT.ELEMENT-PRI.DIR(A))
LET MT=6./PI.C
LET BT=3./(8.*PI.C)
LET AT=1./(2.*PI.C)
LET P.SUB.K=(BT/MT)+AT*(SIN.F(ANGLE+(1/MT))
-SIN.F(ANGLE))
IF P.SUB.K LT 0.0
LET P.SUB.K=0.0
ALWAYS
'CONTINUE'
''IF OBSERVER LOOKING IN THE WRONG DIRECTION RETURN
IF X GT P.SUB.K
LET DET.TIME=99.
GO TO OUT
```

```
ELSE
LET P.HAT=PI.HAT(B)
LET X.VELOCITY=ABS.F(SPD(B)*SIN.F(TGT.ELEMENT-
DIR.OF.MVMT(B)))
GO TO ONE,TWO,THREE PER P.HAT
'ONE'
LET TC.FACTOR=0.53
JUMP AHEAD
'TWO'
LET TC.FACTOR=1.49
JUMP AHEAD
'THREE'
LET TC.FACTOR=1.35
HERE
LET PER.FULL.EXPO=1-CBAR(B)/100.
LET RGE.FACTOR=.018+(.0058*R/PER.FULL.EXPO)
LET CROSSING.VEL.FACTOR=1.39-(.076*X.VELOCITY)
''CALULATE MEAN.DET.TIME
LET MEAN.DET.TIME=1.1*(EXP.F(TC.FACTOR+CROSSING.VEL.
FACTOR+RGE.FACTOR))
IF MEAN.DET.TIME LE 0.0
LET MEAN.DET.TIME=ZL
ALWAYS
LET DET.TIME=LOG.NORMAL.F(MEAN.DET.TIME,0.81,3)
IF DET.TIME LT 0.0 LET DET.TIME=ZL
ALWAYS
```

```
60  'OUT'
61  RETURN
62  END
```

Lines 1-5 define the routine and the local variables.

Lines 6-11 define an upper and lower bound on the detection time based on range to the target.

Lines 12-15 establish a zero limit and set the dummy variable DD to the value of PCT.VIS.

Line 16 sets the variable TGT.ELEMENT equal to the value of the angle from the observer to the target as measured from east in a counter-clockwise direction.

Lines 17-19 transfer control to the label FULL.CARDIOID if the searcher is currently using a 360 degree cardioid.

Lines 20-21 are used when the observer is not looking in a 360 degree cardioid. This implies that the searcher is looking in a sector which is a multiple of 30. Therefore, the probability that the observer is looking in the k-th subinterval is 30/area.

Lines 22-30 are executed if the full 360 degree cardioid is being used by the observer at this time. They determine the probability that the observer is looking in the subinterval where the target is.

Lines 31-36 are used when the observer is looking in the wrong direction. The DET.TIME is then set to 99 which will prevent the detection from taking place.

Line 37 accesses the terrain complexity factor.

Line 38 determines the crossing velocity using the angle from the observer to the target, the speed of the target, and the direction of movement of the target.

Lines 39-48 determine the terrain complexity factor based on the complexity of the terrain around the target.

Lines 52-56 determine the mean detection time of the log-normal detection distribution.

Lines 57-60 determine the detection time from the log-normal distribution.

Lines 61-62 return control to the calling routine.

7. Routine INF.COMPUTE

Description

This routine is used to determine the lethality probabilities for MKILL, FKILL, KKILL, HIT and E.CAS (expected casualties) for selected weapons. These weapons include the VIPER, the LAW, the RPG and the M203DP round. Once the lethality probabilities are accessed, Routine ATRIT is called to attrit the elements.

Local Variables Used in this Routine

E.CAS - A real variable indicating the expected casualty probability for elements riding in a vehicle at the time that a round impacts with the vehicle.

GAMMA - A real variable indicating the size of the angle from the firer's line of sight to the long axis of

the target.

I - An integer variable used to access data according to the system type and weapon type of the target.

J - An integer variable used to access data according to the target disposure.

K - An integer variable used to access data according to the range to the target.

M - An integer variable used to access data according to the orientation angle of the target with respect to the firer.

N - An integer variable indicating the system type of the target element.

NN - An integer variable indicating the weapon type of the target element.

P - An integer variable used as a loop incrementer.

P.FKILL - A real variable indicating the probability of a firepower kill to the target element if he is hit.

P.HIT - A real variable indicating the probability that a firer will hit a given target.

P.KKILL - A real variable indicating the probability that a target will be catastrophically killed if hit by a round.

P.MFKILL - A real variable indicating the probability that a target will suffer a mobility and

firepower kill if he is hit.

P.MKILL - A real variable indicating the probability that a target will be mobility killed if he is hit.

R - A real variable returned from Function DIST indicating the range to the target.

RN - A real variable, which is a UNIFORM number drawn on the interval from 0 to 1.

VEL - A real variable representing the speed at which the target is moving.

WHOCALLED - An integer variable used in the call to Routine ATRIT to indicate that Routine INF.COMPUTE called Routine ATRIT.

X.T - A real computational variable indicating the difference between the X coordinates of the target and the firer.

Y.T - A real computational variable indicating the difference between the Y coordinates of the target and the firer.

Coding and Brief Explanations

```
ROUTINE INF.COMPUTE(SH.T,TGT.T,F.PCVIS,PC.VIS)
DEFINE I,J,K,L,M AS INTEGER VARIABLES
DEFINE N,NN AS INTEGER VARIABLES
DEFINE SH.T,TGT.T AS INTEGER VARIABLES
DEFINE P AS AN INTEGER VARIABLE
DEFINE WHOCALLED AS AN INTEGER VARIABLE
```

```
NORMALLY MODE IS REAL
''USE ROUTINE BRST.FIRE FOR WEAPONS FIRING BURST
IF SYS.TYPE(SH.T)=3 AND WPN.TYPE(SH.T) GE 10 AND
WPN.TYPE(SH.T) NE 12 AND WPN.TYPE(SH.T) NE 13 AND
(PROJO(SH.T)=3 OR PROJO(SH.T)=4))
OR (WPN.TYPE(SH.T)=12 AND PROJO(SH.T)=5)
CALL BRST.FIRE(SH.T,TGT.T,F.PCVIS,PC.VIS,GAMMA)
RETURN
ALWAYS
IF (SYS.TYPE(SH.T)=3 AND WPN.TYPE(SH.T) GE 10 AND
(PROJO(SH.T)=5 OR PROJO(SH.T)=6)) OR
(SYS.TYPE(SH.T)=3 AND WPN.TYPE(SH.T)=12 AND PROJO(SH.T)=3)
CALL FRAG.ROUND(SH.T,TGT.T,R)
RETURN
ALWAYS
IF SYS.TYPE(SH.T)=13 AND PROJO(SH.T)=1 ''RPG FIRING AKMS
CALL BRST.FIRE(SH.T,TGT.T,F.PC.VIS,GAMMA)
RETURN  ALWAYS
SUBSTITUTE THIS LINE FOR RANDOM.NUMBERS
NORMAL.F(0.,1.,6)
SUBSTITUTE THESE 12 LINES FOR ATTRITION.OF.ELEMENTS
FOR P=1 TO DIM.F(RIDER(NAME(SECLDR(TGT.T)),*)),DO
LET RN=UNIFORM.F(0.,1.,1)
IF RN GT E.CAS  CYCLE
ELSE
LET TANK=RIDER(NAME(SECLDR(TGT.T)),P)
```

```
CALL ATRIT(SH.T,TANK,0.,0.,1.1,WHOCALLED)
LOOP
ALWAYS
CALL ATRIT(SH.T,TGT.T,P.MKILL,P.FKILL,P.KKILL,WHOCALLED)
ELSE LET DAMAGE.NUM=6
ALWAYS
RETURN
SUBSTITUTE THESE 8 LINES FOR TGT.DISPOSURE.SET
IF VEL GT 0.0 LET J=1
ELSE
IF(PC.VIS-TARDIM(N,NN,10)) LT (.5*(1-TARDIM(N,NN,10)))
LET J=2  ''STATIONARY/HULL DEFILADE
ELSE
LET J=3  ''STATIONARY/FULLY EXPOSED
ALWAYS
ALWAYS
LET WHOCALLED=3  ''ROUTINE ATRIT IS CALLED BY  INFC
LET R=DIST(SH.T,TGT.T)
LET VEL=SPD(TGT.T)
LET RN=UNIFORM.F(0.,1.,3)
LET N=SYS.TYPE(TGT.T)  LET NN=WPN.TYPE(TGT.T)
''DETERMINE THE TARGET ORIENTATION
LET X.T=X.CURRENT(TGT.T)-X.CURRENT(SH.T)
LET Y.T=Y.CURRENT(TGT.T)-Y.CURRENT(SH.T)
LET GAMMA=ARCTAN.F(Y.T,X.T)
LET GAMMA=ABS.F(PI.C-ABS.F(DIR.OF.MVMT(TGT.T)-GAMMA))
```

```
''GAMMA NOW REPRESENTS THE ANGLE FROM THE FIRER'S LINE
OF SIGHT TO THE TARGET LONG AXIS
''
''VIPER
''
IF SYS.TYPE(SH.T)=3 AND WPN.TYPE(SH.T) GE 10 AND
WPN.TYPE(SH.T) LE 12 AND PROJO(SH.T)=2
''DETERMINE THE TARGET TYPE
IF SYS.TYPE(TGT.T)=1 AND WPN.TYPE(TGT.T)=7 ''T72
LET I=1
ALWAYS
IF SYS.TYPE(TGT.T)=2 AND WPN.TYPE(TGT.T)=8 ''BMP
LET I=2
ALWAYS
''DETERMINE THE TARGET DISPOSURE
TGT.DISPOSURE.SET
''DETERMINE THE RANGE INDEX
LET K=TRUNC.F(R/100.)+1.
''ACCESS THE PROBABILITIES
LET P.HIT=VIPERHEAT(I,J,K,5)/1000.
IF RN LE P.HIT
LET P.MKILL=VIPERHEAT(I,J,K,1)/1000.
LET P.FKILL=VIPERHEAT(I,J,K,2)/1000.
LET P.MFKILL=VIPERHEAT(I,J,K,3)/1000.
LET P.KKILL=VIPERHEAT(I,J,K,4)/1000.
LET E.CAS=VIPERHEAT(I,J,K,6)/1000.
```

```
IF SYS.TYPE(TGT.T)=2 AND WPN.TYPE(TGT.T)=8 AND
   CARR(TGT.T) GT 0
ATTRITION.OF.ELEMENTS
ELSE
''
''RPG
''
IF SYS.TYPE(SH.T) AND WPN.TYPE(SH.T)=13
   AND PROJO(SH.T)=3
''CALCULATE THE TARGET TYPE
IF SYS.TYPE(TGT.T)=1 AND (WPN.TYPE(TGT.T)=1 OR
   WPN.TYPE(TGT.T)=2)  ''XM1
LET I=1
ALWAYS
IF SYS.TYPE(TGT.T)=2 AND WPN.TYPE(TGT.T)=3  ''IFV
LET I=2
ALWAYS
IF SYS.TYPE(TGT.T)=2 AND WPN.TYPE(TGT.T)=4  ''ITV
LET I=3
ALWAYS
''DETERMINE THE TARGET DISPOSURE INDEX
TGT.DISPOSURE.SET
''DETERMINE THE RANGE INDEX
LET K=TRUNC.F(R/100.)+1.
''CHECK TO SEE IF THE TARGET WAS HIT
''ACCESS THE PROBABILITIES
```

```
LET P.HIT=RPGHEAT(I,J,K,5)/1000.
IF RN LE P.HIT
LET P.MKILL=RPGHEAT(I,J,K,1)/1000.
LET P.FKILL=RPGHEAT(I,J,K,2)/1000.
LET P.MFKILL=RPGHEAT(I,J,K,3)/1000.
LET P.KKILL=RPGHEAT(I,J,K,4)/1000.
LET E.CAS=RPGHEAT(I,J,K,6)/1000.
IF SYS.TYPE(TGT.T)=2 AND WPN.TYPE(TGT.T)=3 AND
CARR(TGT.T) GT 0  ''IFV
ATTRITION.OF.ELEMENTS
ELSE
''
''LAW
''
IF SYS.TYPE(SH.T)=3 AND WPN.TYPE(SH.T) GE 10 AND
WPN.TYPE(SH.T) LE 12 AND PROJO(SH.T)=1
''CALCULATE THE TARGET TYPE
IF SYS.TYPE(TGT.T)=1 AND WPN.TYPE(TGT.T)=7  ''T72
LET I=1
ALWAYS
IF SYS.TYPE(TGT.T)=2 AND WPN.TYPE(TGT.T)=8  ''BMP
LET I=2
ALWAYS
''DETERMINE THE SPEED
IF VEL LT .01
LET J=1
```

```
ELSE
IF VEL LE 10.
LET J=2
ELSE
LET J=3
ALWAYS
ALWAYS
''DETERMINE THE RANGE INDEX
LET K=TRUNC.F(R/50.)+1
''DETERMINE THE DEFILADE DISPOSURE
IF(PC.VIS-TARDIM(N,NN,10)) GE (.5*(1-TARDIM(N,NN,10)))
LET L=1  ''FULLY EXPOSED
ELSE
LET L=2  ''HULL DEFILADE
ALWAYS
''DETERMINE THE ORIENTATION ANGLE
LET M=TRUNC.F(GAMMA/30.)+1
IF M GE 6  LET M=6
ALWAYS
''ACCESS THE PROBABILITIES
LET P.HIT=LAWHEAT(I,J,K,L,M,5)/100.
IF RN LE P.HIT
LET P.KKILL=LAWHEAT(I,J,K,L,M,1)/100.
LET P.MKILL=LAWHEAT(I,J,K,L,M,2)/100.
LET P.FKILL=LAWHEAT(I,J,K,L,M,3)/100.
LET P.MFKILL=LAWHEAT(I,J,K,L,M,4)/100.
```

```
LET E.CAS=LAWHEAT(I,J,K,L,M,6)/100.
IF SYS.TYPE(TGT.T)=2 AND WPN.TYPE(TGT.T)=8 AND
CARR(TGT.T) GT 0
ATTRITION.OF.ELEMENTS
ELSE
''
''M203DP
''
IF SYS.TYPE(SH.T)=3 AND WPN.TYPE(SH.T)=12 AND
PROJO(SH.T)=4
''DETERMINE THE TARGET TYPE FOR DP ROUND
IF SYS.TYPE(TGT.T)=2 AND WPN.TYPE(TGT.T)=8  ''BMP
LET I=1
ALWAYS
IF SYS.TYPE(TGT.T)=2 AND WPN.TYPE(TGT.T)=-99
''IF NEEDED A BRDM CAN BE ASSESSED HERE BY CHANGING
-99 TO AN APPROPRIATE USER DEFINED WEAPON TYPE
FOR A BRDM
LET I=2
ALWAYS
''DETERMINE THE TARGET DISPOSURE
TGT.DISPOSURE.SET
''DETERMINE THE RANGE INDEX TO THE TARGET
LET K=TRUNC.F(R/50.)+1
""ACCESS THE PROBABILITIES
LET P.HIT=M203DP(I,J,K,5)/100.
```

```
179 IF RN LE P.HIT
180 LET P.MKILL=M203DP(I,J,K,1)/100.
181 LET P.FKILL=M203DP(I,J,K,2)/100.
182 LET P.MFKILL=M203DP(I,J,K,3)/100.
183 LET P.KKILL= M203DP(I,J,K,4)/100.
184 ATTRITION.OF.ELEMENTS
185 ALWAYS
186 RETURN
187 END
```

Lines 1-7 define the routine and the local variables.

Lines 8-14 transfer which can fire bursts to Routine BRST.FIRE to be assessed.

Lines 15-20 transfer weapons which produce fragments to Routine FRAG.ROUND to be assessed.

Lines 21-49 set up substitute names for blocks of repetitious coding. The value for the variable WHOCALLED is also set.

Lines 50-53 determine the distance to the target, the velocity of the target, the system type of the target, the weapon type of the target and a Uniform random number.

Lines 54-60 determines the angle from the firer's line of sight to the target long axis.

Lines 61-64 screen the firer to see if he is firing a Viper.

Lines 65-71 determine the type of target and sets the appropriate index.

Lines 72-73 classify the target according to whether or not

it is moving and, if it is stationary, what its defilade status is. Notice that the substitute TGT.DISPOSURE.SET actually stands for 8 lines of coding.

Lines 74-75 determine the range index for accessing the data.

Lines 76-83 access the accuracy and lethality probabilities.

Lines 84-86 attrit the elements in the target vehicle according to the expected number of casualties.

Lines 87-91 screen the firer to determine if it is an RPG gunner.

Lines 92-100 determine the target type and the appropriate index.

Lines 101-102 determine the index for the target disposure.

Lines 103-104 determine the range index for the distance to the target.

Lines 105-113 access the accuracy and lethality probabilities for the RPG.

Lines 114-116 attrit the vehicle and its riders for the expected casualties resulting from the round's impact.

Lines 117-161 set the appropriate indices, access the accuracy and lethality data, access the expected casualty data and apply the attrition to the target elements hit by a LAW.

Lines 162-190 set the appropriate indices, access the accuracy and lethality data, access the expected casualty data and apply the appropriate attrition to the target

elements hit by an M203DP round.

8. Routine BRST.FIRE

Description

This routine accepts as inputs the pointer variables for the firer, the target, the percent visible of the target at fire and the percent of the target visible at impact. The firer, by virtue of a call to this routine, is firing a weapon capable of burst fire. The effectiveness of the weapon is degraded for the suppressive effects from rounds impacting in the vicinity of the firer. The apparent size of the target is calculated and the expected burst size is accessed from Array BRST. Each round is tracked to the target and a determination is made as to whether the projectile hit the target. If the round hit the target, the user specified input determines whether or not detailed casualty assessment is played. If so, Routine INCAPACITATE is called to determine the body part hit and the resulting damage to the target. If the user does not desire detailed casualty assessment, the probability of kill given hit is 1.

Local Variables Used in This Routine

AIMERROR - A real variable indicating the aim error. This is measured in mils.

AIMDIS - A real variable representing the distance down from the top of the target that the firer aimed. This is a positive number.

BALLERR - A real variable indicating the round to round ballistical dispersion measured in mils.

BRSTSIZE - A real variable used to calculate the actual burst size that the firer will fire.

BSIZE - An integer variable indicating the total number of rounds in the burst that the firer will fire.

CIPDEFL - A real variable representing the deflection distance from the aim point to the center of impact of the subsequent rounds.

CIPEL - A real variable indicating the elevation distance from the aim point to the center of impact of the subsequent rounds.

DEFLMISS - A real variable which represents the absolute value of the total deflection miss distance by which the firer missed his aim point.

DEFSIG1 - A real variable representing the standard deviation of the deflection distance from the first round to the center of impact of the subsequent rounds in the burst.

DMISS - A real variable which is equal to the "signed" value of DEFLMISS. That is, the total deflection miss distance by which the firer missed his target on the first round.

DSIG2 - A real variable representing the standard deviation of the deflection of the subsequent rounds in a burst about their center of impact.

DUMMY - A real variable representing the user defined expected burst size of the firing weapon.

ELEVSIG1 - A real variable representing the standard deviation of the elevation distance from the first round to the center of impact of the subsequent rounds in the burst.

ELMISS - A real variable which represents the total elevation miss distance by which the firer missed his aim point on a given round.

EMISS - A real variable which indicates the total elevation miss distance by which the firer missed his aim point on the first round.

ESIG2 - A real variable representing the standard deviation of the elevation dispersion of the subsequent projectiles about their center of impact.

GAMMA - A real variable representing the angle from the firer's line of sight to the target long axis.

HORIZDIS - A real variable representing the deflection distance in mils from the first round to the center of impact of the subsequent rounds.

I - An integer variable used in accessing accuracy data.

J - An integer variable used in accessing accuracy

data.

K - An integer variable used in accessing accuracy data.

L - An integer variable used in accessing accuracy data.

LENGTH - A real variable accessed from Array TARDIM representing the size of the target from front to rear (i.e. width of the target as viewed from the side).

M.SX - A real variable used to adjust the firer's deflection miss distance for suppression.

M.SY - A real variable used to adjust the firer's elevation miss distance for suppression.

N - An integer variable representing the system type of the target.

N.RDS - An integer variable indicating the round number of the current round in the burst being evaluated.

NN - An integer variable representing the weapon type of the target.

RD.STD.DEV - A real variable indicating the first round common error. It is composed of the aim error and the round to round dispersion.

SIZE - A real variable representing the apparent width of the torso of the target.

SIZE.HEAD - A real variable representing the apparent width of the head of the target.

VERTDIS - A real variable representing the elevation distance from the first round to the center of impact of the subsequent rounds.

WHOCALLED - An integer variable used in debugging. This variable indicates that Routine BRST.FIRE called Routine ATRIT.

WIDTH - A real variable accessed from array TARDIM representing the width of the target as viewed from the front.

Coding and Brief Explanation

```
ROUTINE BRST.FIRE(SH.T,TGT.T,F.PCVIS,PC.VIS,GAMMA)
DEFINE AIMDIS AS A REAL VARIABLE
DEFINE N.RDS AS AN INTEGER VARIABLE
DEFINE AIMBIAS,BALLERR,R,BRSTSIZE,CIPDEFL,CIPEL,DEFLMISS,
DEFLSIG1,DMISS,DSIG2,DUMMY,ELEVSIG1,ELMISS
EMISS,ESIG2,GAMMA
HORIZDIS,LENGTH,M.SX,M.SY,F.PC.VIS,STD.DEV,SIZE,
SIZE.HEAD,VERTDIS,WIDTH AS REAL VARIABLES
DEFINE BSIZE,WHOCALLED AS INTEGER VARIABLES
DEFINE P AS AN INTEGER VARIABLE
DEFINE N,NN,I,J,K,L AS INTEGER VARIABLES
DEFINE SH.T,TGT.T AS INTEGER VARIABLES
LET R=DIST(SH.T,TGT.T)
LET N=SYS.TYPE(TGT.T)  LET NN=WPN.TYPE(TGT.T)
SUBSTITUTE THIS LINE FOR RANDOM.NUMBERS
```

```
NORMAL.F(0.,1.,6)
''NOW DEAL WITH SUPPRESSION
LET M.SX=1.  LET M.SY=1.
''SUPPRESSION CHECK
IF TIM.SP(SH.T) GE GET.SP(SH.T,7,0)
LET M.SX=GET.SP(SH.T,8,0)
LET M.SY=GET.SP(SH.T,9,0)
ALWAYS
''AIM POINT IS THE CENTER MASS OF THE TARGET
LET AIMDIS=F.PC.VIS*TARDIM(N,NN,4)/2.
''AIMDIS IS NOW A POSITIVE NUMBER REPRESENTING
''A DISTANCE DOWN FROM THE TOP OF THE TARGET THAT
''THE FIRER AIMED
''CALCULATE THE APPARENT WIDTH OF THE TARGET
IF PC.VIS GT TARDIM(N,NN,10)  ''IF MORE THAN THE
HEAD SHOWS, USE THE WIDTH OF THE BODY
LET WIDTH=TARDIM(N,NN,6)
LET LENGTH=TARDIM(N,NN,7)
ELSE
''WIDTH OF THE HEAD IS USED
LET WIDTH=TARDIM(N,NN,5)
LET LENGTH=TARDIM(N,NN,8)
ALWAYS
LET SIZE=ABS.F(LENGTH*SIN.F(GAMMA))
+ABS.F(WIDTH*COS.F(GAMMA))
''DETERMINE THE RANGE INDEX
```

```
IF R LT 50.  LET K=1
JUMP AHEAD
ALWAYS
IF R LT 100. LET K=2 JUMP AHEAD ELSE
LET K=TRUNC.F(R/100.)+2
IF K GT 7  LET K=7  ALWAYS
HERE
''CALCULATE THE BURST SIZE
LET DUMMY=BRST(SYS.TYPE(SH.T),
(WPN.TYPE(SH.T)-9),PROJO(SH.T))
LET BRSTSIZE=POISSON.F(DUMMY,1)
IF BRSTSIZE EQ 0 LET BRSTSIZE=1 ALWAYS
''M16 OR M16 PART OF M203
''
IF WPN.TYPE(SH.T)=10 OR WPN.TYPE(SH.T)=12
LET I=2  LET J=4  LET L=3
IF SYS.TYPE(SH.T)=3 AND WPN.TYPE(SH.T)=10
AND PROJO(SH.T)=3
''SUBJECT THE BURST SIZE TO AMMO CONSTRAINT
LET BSIZE=INT.F(MIN.F(BRSTSIZE,AMMO3(SH.T)))
ALWAYS
IF(WPN.TYPENSH.T)=10 AND PROJO(SH.T)=4) OR
(WPN.TYPE(SH.T)=12 AND PROJO(SH.T)=6)
LET BSIZE=MIN.F(1,AMMO4(SH.T))
GO TO PROB.EVAL
ALWAYS
```

```
ALWAYS
''
''M60
''
IF WPN.TYPE(SH.T)=11 AND PROJO(SH.T)=2
LET I=1 LET J=1  LET L=1
''SUBJECT BURST TO AMMO CONSTRAINT
LET BSIZE=INT.F(MIN.F(BRSTSIZE,AMMO2(SH.T)))
GO TO PROB.EVAL
ALWAYS
''SAW
IF WPN.TYPE(SH.T)=11 AND PROJO(SH.T)=3
LET I=1  LET J=3  LET L=2
''SUBJECT BURST TO AMMO CONSTRAINT
LET BSIZE=INT.F(MIN.F(BRSTSIZE,AMMO3(SH.T)))
GO TO PROB.EVAL
ALWAYS
''
AKMS
''
IF WPN.TYPE(SH.T)=14
LET I=3 LET J=5  LET L=4
''SUBJECT BURST TO AMMO CONSTRAINT
LET BSIZE=INT.F(MIN.F(BRSTSIZE,AMMO3(SH.T)))
GO TO PROB.EVAL
ALWAYS
```

```
''
PKM
''
IF WPN.TYPE(SH.T)=15
LET I=2  LET J=2  LET L=1
''SUBJECT BURST TO AMMO CONSTRAINT
LET BSIZE=INT.F(MIN.F(BRSTSIZE,AMMO3(SH.T)))
GO TO PROB.EVAL
ALWAYS
'PROB.EVAL'
''ACCESS ERRORS
LET AIMBIAS=AIMERROR(L,K)/100.
LET BALLERR=.4  ''UNCLASSIFIED DATA SOURCE
''DETERMINE THE STD DEV OF THE PROJECTILES
LET RD.STD.DEV=SQRT.F(AIMBIAS**2+BALLERR**2)
''DETERMINE THE HORIZONTAL AND VERTICAL DISTANCE
''FROM THE FIRST ROUND TO THE CENTER OF IMPACT
''OF THE SUBSEQUENT ROUNDS
LET HORIZDIS=DISPERSIONS(I,J,1)/10.
LET VERTDIS=DISPERSIONS(I,J,2)/10.
''DETERMINE SIGX AND SIGY FOR THE DISTANCE FROM
''THE IMPACT OF THE FIRST ROUND TO THE CENTER
''OF IMPACT OF THE SUBSEQUENT ROUNDS
LET DEFLSIG1=DISPERSIONS(I,J,3)/10.
LET ELEVSIG1=DISPERSIONS(I,J,4)/10.
''DETERMINE SIGX,SIGY FOR THE DISPERSION OF
```

```
''THE SUBSEQUENT ROUNDS ABOUT THEIR CENTER OF
''IMPACT
LET DSIG2=DISPERSIONS(I,J,5)/10.
LET ESIG2=DISPERSIONS(I,J,6)/10.
'BURST.EVAL'
IF BSIZE EQ 0  RETURN ALWAYS
LET N.RDS=1
''DID THE FIRST ROUND HIT THE TARGET?
LET DEFLMISS=RANDOM.NUMBERS*RD.STD.DEV
LET ELMISS=RANDOM.NUMBERS*RD.STD.DEV
''ACCOUNT FOR SUPPRESSION
LET DEFLMISS=DEFLMISS*M.SX
LET ELMISS=ELMISS*M.SY
LET DMISS=DEFLMISS
LET DEFLMISS=ABS.F(DEFLMISS)
''CONVERT TO METERS
''DMISS HAS MAINTAINED THE SIGN OF DEFLMISS
LET ELMISS=ELMISS*R/1000.
LET DEFLMISS=DEFLMISS*R/1000.
LET EMISS=ELMISS
'HIT.OR.MISS'
''FIND OUT IF THE SHOT IS HIGH OR LOW
IF ELMISS GT AIMDIS  ''MISSED OVER THE TARGET
LET PH(SH.T)=3  LET DAMAGE.NUM=6  JUMP AHEAD
ALWAYS
IF ELMISS GO 0 AND (((AIMDIS-ELMISS)/TARDIM(N,NN,4))
```

```
GT PC.VIS)
''MISSED SHORT IN THE DIRT
GO TO DRT
ALWAYS
IF ELMISS LT 0 AND ((AIMDIS-ELMISS) GT (PC.VIS*
TARDIM(N,NN,4)))
''MISSED SHORT IN THE DIRT
GO TO DRT
ELSE GO TO DEFCHK
'DRT'
CALL WGT.SP(SH.T,0,0,TGT.T)
LET PH(SH.T)=4  LET DAMAGE.NUM=6  JUMP AHEAD
'DEFCHK'
''IF WE GET HERE WE ARE ON IN ELEVATION SO CHECK FOR
''DEFLECTION
IF DEFLMISS GT SIZE/2.  ''MISSED UNSENSED
LET PH(SH.T)=3  LET DAMAGE.NUM=6  JUMP AHEAD
ALWAYS
''IF BULLET WAS HIGH WHEN AIMING AT THE BODY
''CHECK TO SEE IF IT MISSED THE HEAD
IF ELMISS GT 0.0 AND ((AIMDIS-TARDIM(N,NN,2)-ELMISS)
LT 0.0)  ''BULLET IS ABOVE THE SHOULDER
LET WIDTH=TARDIM(N,NN,5)
LET LENGTH=TARDIM(N,NN,8)
LET SIZE.HEAD=ABS.F(LENGTH*SIN.F(GAMMA))+
ABS.F(WIDTH*COS.F(GAMMA))
```

```
IF DEFLMISS GT SIZE.HEAD/2.  ''SHOT WIDE OVR SHLDR
LET PH(SH.T)=3
LET DAMAGE.NUM=6  JUMP AHEAD
ALWAYS
ALWAYS
''CHECK THE SITUATION WHERE THE AIM POINT WAS THE HEAD
''AND THE SHOT WENT OVER THE SHOULDER
IF (AIMDIS-ELMISS)  LT TARDIM(N,NN,2) AND DEFLMISS
GT SIZE.HEAD/2.
LET PH(SH.T)=3
LET DAMAGE.NUM=6  JUMP AHEAD
ALWAYS
''IF WE GET TO HERE THE TARGET WAS HIT
''OPTION OF ALLOWING THE PROBABILITY OF KILL
GIVEN HIT TO EQL 1 OR TO ASSESS DETAILED INCAPAC
''CHECK THE USER INPUT VALUE OF INF.CAS.ASSMT
IF INF.CAS.ASSMT=1
CALL INCAPACITATE(SH.T,TGT.T,DEFLMISS,EMISS,AIMDIS,SIZE,
SIZE.HEAD,R)
ELSE  ''A HIT IS DETERMINED TO BE A KILL
CALL ATRIT(SH.T,TGT.T,0.,0.,1.1,WHOCALLED)
ALWAYS
HERE
'SUBSEQUENT.RDS'
IF N.RDS GT 1 JUMP AHEAD ELSE
''CALCULATE THE CENTER OF IMPACT OF THE SUBSEQUENT ROUNDS
```

```
181  LET CIPDEFL=HORIZDIS+RANDOM.NUMBERS*DEFLSIG1
182  LET CIPEL=VERTDIS+RANDOM.NUMBERS*ELEVSIG1
183  LET CIPDEFL=DMISS*R/1000.+CIPDEFL*R/1000.
184  LET CIPEL=ELMISS+CIPEL*R/1000.
185  HERE
186  LET N.RDS=N.RDS+1
187  IF N.RDS GT BSIZE RETURN  ELSE
188  ''NOW CALCULATE WHERE THE SUBSEQUENT ROUNDS IMPACTED
189  LET DEFLMISS=CIPDEFL+RANDOM.NUMBERS*DSIG2*R/1000.
190  LET ELMISS=CIPEL+RANDOM.NUMBERS*ESIG2*R/1000.
191  IF N.RDS LE BSIZE
192  GO TO HIT.OR.MISS  ''ASSESS THE NEXT ROUND
193  ALWAYS
194  RETURN  END
```

Lines 1-11 define the variables and the routine.

Lines 12-13 determine the range to the target. Additionally the substitute variables N and NN are set.

Lines 14-15 set up a substitute name for a draw for a Normal random number.

Lines 16-22 access the suppression data.

Lines 23-27 set the value for AIMDIS. This represents the positive distance down from the top of the target the firer is aiming.

Lines 28-38 calculate the apparent width of the target. This accounts for increased width due to the target long axis not lying on or perpendicular to the firer's line of

sight line.

Lines 39-45 set the appropriate index for the range to the target. This index will be used to access the accuracy arrays.

Lines 46-48 draw a Poisson random number to assist in the determination of the burst size that the firer will fire if he has sufficient ammunition remaining.

Lines 49-64 determine the weapon type of the firer and the burst size is set to the minimum of the stochastically determined burst size and the remaining ammunition of the firer.

Lines 73-79 do the above described actions for the SAW.

Lines 80-88 do the above described actions for the AKMS.

Lines 89-97 do the above described actions for the PKM.

Lines 98-112 access the appropriate accuracy data arrays.

Lines 113-118 check to see if the firer is out of ammunition. If he is not, the standard deviations for the error of the first round are computed resulting in DEFLMISS and ELMISS. DEFLMISS at this point is the total miss deflection of the first round in mils. ELMISS represents the total miss elevation of the first round in mils.

Lines 119-121 adjust the deflection and elevation misses of the first round to account for the effects of suppression.

Lines 122-123 set computational variations of DEFLMISS for use in further calculations.

Lines 124-127 convert the values of ELMISS and DEFLMISS to

meters. Note that at this point DEFLMISS represents the absolute value of the miss distance of the first round in meters. ELMISS is now the total miss of the first round in meters. Additionally EMISS is set equal to ELMISS for computational reasons.

Lines 128-132 check to see if the shot was higher than the head.

Lines 133-142 conduct a series of checks to see if the shot was low in the dirt.

Lines 142-144 increment the suppression level in WGT.SP if the shot was low in the dirt. Additionally a miss is recorded.

Lines 145-149 checks for deflection error large enough to have caused the firer to miss the target. The round is on in elevation if the program reaches this portion of the code. The first check that has to be made, therefore, is to see if the round's deflection miss distance is greater than half the width of the body.

Lines 150-160 check to see if the round missed the target in the case where the firer was aiming at the body but the bullet went high. The question is whether or not the round hit the head which is done by comparing the deflection miss distance to half the apparent width of the head. If it exceeds the head width, the round was a miss.

Lines 161-168 check the case where the aim point was the head and the shot went above the shoulder. Again the

apparent size of the head is compared to DEFLMISS.

Lines 169-177 determine that the target has been hit and checks the value of INF.CAS.ASSMT to determine if the user desires detailed casualty assessment or desires to play a hit is assessed as a kill on dismounted elements. If INF.CAS.ASSMT=1, Routine INCAPACITATE will be called. Otherwise Routine ATRIT will be called to incapacitate the element.

Lines 178-184 determine the center of impact of the subsequent rounds in a burst if appropriate.

Lines 185-186 determines if the current round being evaluated is the last round in the burst.

Lines 187-194 determine the elevation and deflection miss distances for the subsequent rounds in a burst. The program flow then goes back to HIT.OR.MISS to evaluate the round.

9. <u>Event MAX.WAIT.TIME</u>

Description

Event MAX.WAIT.TIME is scheduled when a unit is moving from one location to a new location. Once it is determined that a unit desires to move all vehicles will remain in place a selected amount of time to allow dismounted elements time to remount their vehicles. The amount of time a vehicle waits is an input parameter of the simulation. At the time this event takes place each vehicle will make one final check to remount soldiers

that are within five meters of the vehicle. Soldiers that are not within five meters will be left behind and are required to move to their new positions on foot.

Local Variables

A - The pointer to the vehicle that is about to move.

Coding and Brief Explanation

```
UPON MAX.WAIT.TIME(A)
DEFINE A AS AN INTEGER VARIABLE
FOR EACH TANK IN SQD.UNIT(SEC(A)) WITH (SYS.TYPE(TANK))
EQUALS 3 AND ALIVE.DEAD(TANK) EQUALS 0), DO
CALL LOC(TANK)
LET R=DIST(A,TANK)
IF R LE 5 AND ALIVE.DEAD(TANK)=0
''IF SOMEONE IS NOT MOUNTED AND WITHIN FIVE METERS
MOUNT HIM
CALL REMOUNT(TANK,1)
CYCLE
ALWAYS
LOOP
LET MV.STATE(A)=1
LET T.SPD(A)=TIME.V
LET DEFNUM(A)=1
SCHEDULE A DF.CHG(A) IN 1.0 UNITS
FOR EACH TANK IN SQD.UNIT(SEC(A)) WITH
SYS.TYPE(TANK) EQUALS 3 AND ALIVE.DEAD(TANK)
```

```
    EQUALS 0, DO
17  LET MV.STATE(TANK)=1
18  LET T.SPD(TANK)=TIME.V
19  LET AREA.START(TANK)=AREA.START(SQDVEH(TANK))
20  LET AREA.END(TANK)=AREA.END(SQDVEH(TANK))
21  LOOP
22  RETURN
23  END
```

Lines 1-2 define the event and the local variables.

Lines 3-11 directs each vehicle to determine if any dismounted soldiers are within a specified distance of the vehicle. If there are soldiers within the specified distance of the vehicle they are allowed to mount the vehicle.

Lines 12-15 set attributes of the vehicle to the necessary values to allow the vehicle to move to a new location.

Lines 16-22 set attributes of those soldiers that did not make it to their squad vehicle to values which will allow each soldier to move back to his next defensive position dismounted.

Line 23 returns control to the system timer.

10. <u>Routine INCAPACITATE</u>

Description

Routine INCAPACITATE is called when the global variable INF.CAS.ASSMT is set to 1. The alternative, when INF.CAS.ASSMT is set equal to 0, is that the probability of kill given a hit is equal to 1. When

INF.CAS.ASSMT equals 1, however, detailed casualty assessment is played. The routine is called from Routine BRST.FIRE with a target element which has been hit by small arms fire where small arms fire also includes the M18 Claymore and the fragments from the M203 grenade launcher. Routine INCAPACITATE determines the part of the target that was hit by the projectile or fragments. The modelled torso for the soldier is diagrammed in Figure 4-1. The process used to determine the body part hit is one of successive elimination. Once the body part hit by the round is determined, Arrays A.CSLTY.DATA, B.CSLTY.DATA, and C.CSLTY.DATA are accessed to retrieve coefficients a, b, and c for a regression used to determine the probability of incapacitation.

The casualty data arrays are indexed by the range to the target, the type of projectile, and the expected time to incapacitation. The first lookup for the coefficients is to determine the probability of incapacitation by time 30 seconds, given that the target element was hit in a specific body part. This is compared to a random number to determine if the incapacitation is to occur by this time. If the random number is larger, a similar check is made for 5 minutes. The final check, if needed, is to see if the target element is incapacitated by time 30 minutes. This

probability of incapacitation is not calculated until the final round in a burst has been fired as signified by the value of the flag variable CALC being set to 1 in Routine BRST.FIRE. For each round prior to the final round, the probability that that particular round caused incapacitation is calculated and stored as an attribute of the target element. Additionally, the number of rounds that have impacted with the specific body part of the entity is incremented. Once the final round is fired, the total body probability of incapacitation is calculated as explained. If the target element is determined to be incapacitated, Routine ATRIT is called to implement the casualty status of the element. If the entity is not determined to have suffered any degradation, he still carries forward some residual incapacitation in the form of the retaining of the number of hits the element has sustained in each body part. This information is used again if the element is hit by another round.

Local Variables Used in This Routine

A - A real coefficient accessed from the Array A.CSLTY.DATA used to determine the probability of incapacitation.

AA - A real variable analogue of A but for the opposite mode of activity from that of the target element.

ABDO.HT - A real variable input by the user indicating the height of the abdomen of a soldier.

B - A real coefficient accessed from Array B.CSLTY.DATA used to determine the probability of incapacitation.

BB - A real variable analogue of B but for the opposite mode of activity from that of the target element.

C - A real coefficient accessed from the Array C.CSLTY.DATA used to determine the probability of incapacitation.

CC - A real variable analogue of C but for the opposite mode of activity from that of the target element.

DEFLMISS - A real argument used to indicate the overall distance in deflection by which the projectile missed the center mass of the target.

ELMISS - A real argument used to indicate the overall distance in elevation by which the projectile missed the center mass of the target.

I - An integer variable used to index the weapon type of the projectile. This is used to access the casualty data arrays.

J - An integer variable used to index the activity of the target element. This is used to access the casualty data arrays.

JJ - An integer variable used to index the opposite mode of activity than that of the target element. This is used to access the casualty data arrays.

L - An integer variable used to index the body part hit by the projectile. This is used to access the casualty data arrays.

N - An integer variable indicating the system type of the target element.

NN - An integer variable indicating the weapon type of the target element.

P.CAS - A real variable indicating the probability that a wound in a given body part will result in incapacitation by a given time.

P.FKILL - A real variable indicating the probability that the target element will suffer a firepower kill as a result of his wound.

P.KKILL - A real variable indicating the probability that the target element will suffer a catastrophic kill as a result of his wound.

P.MFKILL - A real variable indicating the probability that the target element will suffer a mobility and firepower kill as a result of his wound.

P.MKILL - A real variable indicating the probability that the target element will suffer a mobility kill as a result of his wound.

P.ONLY - A real variable indicating the probability

that a given target element will suffer the assumed level of incapacitation only.

PA - A real variable indicating the probability that the target suffers total body incapacitation if he is an attacking element.

PD - A real variable indicating the probability that the target suffers total body incapacitation if he is a defending element.

PP.BODY.IN - A real variable indicating the probability of total body incapacitation for the target element.

PP.CAS - A real variable indicating the probability that a wound in a given body part will result in incapacitation by a given time for the opposite mode of activity of the target element.

R - A real argument indicating the range from the firer to the target. For fragmenting rounds this indicates the distance from the detonation of the round to the target.

RN - A real random number used to determine whether or not a target element is incapacitated.

SET - An integer variable indicating the category that the target element falls into based on his wounds and his mode of activity.

SH.T - An integer variable indicating the pointer value of the firing element.

SIZE - A real variable indicating the apparent width of the body of the target.

SIZE.HEAD - A real variable indicating the apparent width of the head.

TF - A real proportionality variable used in calculating the probability of incapacitation.

TGT.T - An integer variable indicating the pointer value of the target element.

TO - A real proportionality variable used in calculating the probability of incapacitation.

Coding and Brief Explanation

```
ROUTINE INCAPACITATE(SH.T,TGT.T,DEFLMISS,ELMISS,AIMDIS,
SIZE,SIZE.HEAD,R,CALC)
DEFINE JJ AS AN INTEGER VARIABLE
DEFINE A,B,C,DEFLMISS,ELMISS,P.CAS,R,RN,SIZE,
SIZE.HEAD AS REAL VARIABLES
DEFINE SET AS AN INTEGER VARIABLE
NORMALLY MODE IS REAL
DEFINE I,J,K,L,N,NN,SH.T,TGT.T AS INTEGER VARIABLES
''IF THIS IS LAST ROUND IN BURST GO TO CALCULATE
IF CALC=1  GO TO CALCULATE  ALWAYS
LET K=1
''DETERMINE THE WEAPON INDEX
LET I=WPN.TYPE(SH.T)-8
IF SYS.TYPE(SH.T)=3 AND WPN.TYPE(SH.T)=11 AND
```

```
PROJO(SH.T)=4  ''SAW
LET I=1
ALWAYS
IF SYS.TYPE(SH.T)=3 AND WPN.TYPE(SH.T) GE 10
AND WPN.TYPE(SH.T) LE 12 AND PROJO(SH.T)=5  ''CLAYMORE
LET I=5
ALWAYS
''DETERMINE THE ROLE OF THE WOUNDED SOLDIER AT
THE TIME HE WAS SHOT
IF COLOR(TGT.T)=RED
LET J=1  ''ASSAULT
LET JJ=2
ELSE
LET J=2  ''DEFENSE
LET JJ=1
ALWAYS
''DETERMINE THE PART OF THE BODY THAT WAS HIT
LET N=SYS.TYPE(TGT.T)  LET NN=WPN.TYPE(TGT.T)
''DETERMINE IF THE BULLET HIT IN THE BODY OR
''OR IN THE HEAD
''AIMED AT THE BODY AND THE BULLET WAS LOW
IF AIMDIS GT TARDIM(N,NN,2) AND ELMISS LT 0.0
GO TO BODY
ALWAYS
''AIMED AT THE BODY, THE BULLET WAS HIGH BUT
''STILL ON THE BODY
```

```
IF AIMDIS GT TARDIM(N,NN,2) AND (AIMDIS-ELMISS)
GT TARDIM(N,NN,2)
GO TO BODY
ALWAYS
IF ELMISS LT 0.0 AND (AIMDIS+ABS.F(ELMISS)) GT
TARDIM(N,NN,2)
GO TO BODY
ALWAYS
''BULLET IS IN THE HEAD DETERMINE THE DAMAGE INDEX
LET L=1  ''HIT IN THE HEAD
LET N.HD(TGT.T)=N.HD(TGT.T)+1
GO TO DAMAGE.EVAL
'BODY'
IF AIMDIS-ELMISS GT (TARDIM(N,NN,2)+THORAX.HT+
ABDO.HT)
LET L=4  ''PELVIS AND LOWER EXTREMITIES
LET N.PLE(TGT.T)=N.PLE(TGT.T)+1
GO TO DAMAGE.EVAL
ALWAYS
''CHECK FOR THE ARMS
IF ABS.F(DEFLMISS) GT AB.TH.WIDTH/2.
LET L=5  ''HIT IN THE UPPER EXTREMITIES
LET N.ARM(TGT.T)=N.ARM(TGT.T)+1
GO TO DAMAGE.EVAL
ALWAYS
''CHECK FOR ABDOMEN
```

```
58  IF (AIMDIS-ELMISS-TARDIM(N,NN,2) -THORAX.HT)
    GT 0.0
59  LET L=3  ''HIT IN THE ABDOMEN
60  LET N.AB(TGT.T)=N.AB(TGT.T)+1
61  ELSE
62  ''CHECK THE THORAX
63  LET L+2  ''HIT IN THE THORAX
64  LET N.TH(TGT.T)=N.TH(TGT.T)+1
65  ALWAYS
66  'DAMAGE.EVAL'
67  ''DETERMINE THE INCAPACITATION PROBABILITIES
68  LET A=A.CSLTY.DATA(I,J,K,L)/(-10000.)
69  LET AA=A.CSLTY.DATA(I,JJ,K,L)/(-10000.)
70  LET B=B.CSLTY.DATA(I,J,K,L)/(-1000000.)
71  LET BB=B.CSLTY.DATA(I,JJ,K,L)/(-1000000.)
72  LET C=C.CSLTY.DATA(I,J,K,L)/1000000000.
73  LET CC=C.CSLTY.DATA(I,JJ,K,L)/1000000000.
74  IF WPN.TYPE(SH.T)=12  LET C=C*1000.  LET CC=CC*1000.
    ALWAYS  ''DATA PACKING CORRECTION
75  LET P.CAS=EXP.F(A+B*R+C*R*R)
76  LET PP.CAS=EXP.F(AA+BB*R+CC*R*R)
77 GO TO HEAD,TH,AB,PLE,ARMS PER L
78  'HEAD'
79  LET H.IN=PP.CAS
80  LET P.HD.INCAP(TGT.T)=P.CAS
81  JUMP AHEAD
```

```
'TH'
LET TH.IN=PP.CAS
LET P.TH.INCAP(TGT.T)=P.CAS
JUMP AHEAD
'AB'
LET AB.IN=PP.CAS
LET P.AB.INCAP(TGT.T)=P.CAW
JUMP AHEAD
'PLE'
LET PLE.IN=PP.CAS
LET P.PLE.INCAP(TGT.T)=P.CAS
JUMP AHEAD
'ARMS'
LET P.ARM.INCAP(TGT.T)=P.CAS
LET ARM.IN=PP.CAS
HERE
'CALCULATE'
IF CALC=0  RETURN  ELSE  ''LAST ROUND IN THE
''BURST HAS BEEN FIRED
LET P.BODY.INCAP(TGT.T)=1-((1-P.HD.INCAP(TGT.T)**
N.HD(TGT.T))*((1-P.TH.INCAP(TGT.T))**N.TH(TGT.T))*
((1-P.AB.INCAP(TGT.T))**N.AB(TGT.T))
*((1-P.PLE.INCAP(TGT.T))
**N.PLE(TGT.T))*((1-P.ARM.INCAP(TGT.T))**N.ARM(TGT.T))
LET PP.BODY.IN=1-((1-H.IN)**N.HD(TGT.T))*(1-AB.IN)**
N.AB(TGT.T))*((1-TH.IN)**N.TH(TGT.T))*((1-PLE.IN)**
```

```
N.PLE(TGT.T))*((1-ARM.IN)**N.ARM(TGT.T))
IF COLOR(TGT.T)=BLUE
LET PD=P.BODY.INCAP(TGT.T) LET PA=PP.BODY.IN
ELSE
LET PA=P.BODY.INCAP(TGT.T) LET PD=PP.BODY.IN
ALWAYS
LET RN=UNIFORM.F(0.,1.,1)
IF P.BODY.INCAP(TGT.T) LE RN  ''SOLDIER NOT
''INCAPACITATED BY THIS TIME
LET K=K+1
IF K GT 3 RETURN  ELSE
GO TO DAMAGE.EVAL
ELSE
IF K=1  LET TIM.IN=UNIFORM.F(0.,30.,1)  ALWAYS
IF K=2  LET TIM.IN=UNIFORM.F(30.,300.,1)  ALWAYS
IF K=3  LET TIM.IN=UNIFORM.F(300.,1800.,1)  ALWAYS
''TARGET IS INCAPACITATED TO SOME EXTENT, DETERMINE
''THE EXTENT
''FIRST DETERMINE THE APPROPRIATE SET OF THE CASUALTY
''SET 1--TARGET IS HIT IN THE ARMS ONLY OR IS A
''DEFENDER WHO IS HIT IN THE HEAD,THORAX,ABDOMEN,
''OR UPPER AND LOWER EXTREMITIES
IF N.ARM(TGT.T) GE 1 AND (N.HD(TGT.T)+N.AB(TGT.T)+
N.TH(TGT.T)+N.PLE(TGT.T)=0)
OR COLOR(TGT.T)=BLUE AND (N.HD(TGT.T)+N.TH(TGT.T)+
N.AB(TGT.T) GT 0.0) OR (N.ARM(TGT.T)
```

```
    NE 0 AND N.PLE(TGT.T) NE 0.0))
128 LET SET=1
129 JUMP AHEAD
130 ALWAYS
131 ''SET 2--THE TARGET IS A DEFENDER ELEMENT WHO IS HIT
    ''IN THE LEGS ONLY
132 IF COLOR(TGT.T)=BLUE AND N.PLE(TGT.T) GE 1 AND
133 (N.AB(TGT.T)+N.TH(TGT.T)+N.HD(TGT.T)+N.ARM(TGT.T)=0.0)
134 LET SET=2
135 JUMP AHEAD
136 ALWAYS
137 ''SET 3--THE TARGET IS AN ATTACKER ELEMENT WHO HAS WOUNDS
    ''ANYWHERE IN THE BODY EXCEPT IN THE LEGS ONLY
    ''OR IN THE ARMS ONLY
138 IF COLOR(TGT.T)=RED AND  ((N.AB(TGT.T)+
    N.TH(TGT.T)+N.HD(TGT.T)
    +N.ARM(TGT.T) NE 0) AND (N.AB(TGT.T)+
139 N.TH(TGT.T)+N.HD(TGT.T)
140 +N.PLE(TGT.T) NE 0))
141 LET SET=3
142 JUMP AHEAD
143 ALWAYS
144 ''SET 4-THE TGT ELEMENT IS AN ATTACKER HIT
    ''IN THE LEGS ONLY
145 IF COLOR(TGT.T)=RED AND N.PLE(TGT.T) GE 0 AND
    (N.AB(TGT.T)+N.TH(TGT.T)+N.HD(TGT.T)+N.ARM(TGT.T)=0)
```

```
LET SET=4
JUMP AHEAD
ALWAYS
HERE
''DETERMINE THE INCAPACITATION PROBABILITIES
''DETERMINE THE PROPORTIONALITY FACTORS TF AND TO
LET TF=.16*((PA-.5)**2)
LET TO=1.-.2*PA
GO TO ONE,TWO,THREE,FOUR PER SET
'ONE'
LET P.FKILL=1  ''SET 1 INITIALIZED TO FKILL
LET P.ONLY=1-TF  ''PROB OF FKILL ONLY
LET RN=UNIFORM.F(0.,1.,1)
IF RN GT P.ONLY
CALL ATRIT(SH.T,TGT.T,0.,1.,0.,10)
RETURN
ELSE
LET P.MFKILL=TF*(1-TO)
LET P.KKILL=TF*TO
CALL ATRIT(SH.T,TGT.T,P.MFKILL,P.FKILL,P.KKILL,10)
RETURN
'TWO'
LET P.ONLY=1-TO
LET RN=UNIFORM.F(0.,1.,1)
IF RN GT P.ONLY
CALL ATRIT(SH.T,TGT.T,1.,1.,0.,10)
```

```
172 RETURN
173 ELSE
174 LET P.KKILL=TO
175 CALL ATRIT(SH.T,TGT.T,1.,1.,P.KKILL,10)
176 RETURN
177 'THREE'
178 LET P.MKILL=(PA-PD)/PA
179 LET P.FKILL=PD*(1-TF)/PA
180 LET P.MFKILL=TF*PD*(1-TO)/PA
181 LET P.KKILL=TO*TF*PD/PA
182 CALL ATRIT(SH.T,TGT.T,P.MKILL,P.FKILL,P.KKILL,10)
183 RETURN
184 'FOUR'
185 LET P.MKILL=1  ''NORMALIZE FOR AT LEAST MOBILITY KILL
186 LET P.ONLY=(PA-PD)/(PA-PD*(1-TF))
187 LET RN=UNIFORM.F(0.,1.,1)
188 IF RN GT P.ONLY
189 CALL ATRIT(SH.T,TGT.T,P.MKILL,0.,0.,10)
190 RETURN
191 ELSE
192 LET P.FKILL=TF*PD*(1-TO)/(PA-PD*(1-TF))
193 LET P.KKILL=TO*TF*PD/(PA-PD*(1-TF))
194 CALL ATRIT(SH.T,TGT.T,P.MKILL,P.FKILL,P.KKILL,10)
195 RETURN
196 END
```

Lines 1-8 define the routine and the local variables.

Line 9 determines that the final round has been fired, so the total body probability of incapacitation can be calculated.

Lines 10-18 determine the appropriate weapon index of the firer.

Lines 19-26 determine the mode of activity the target was in when he was shot. This is either assault or defense.

Lines 27-40 determine whether the round hit the body or the head of the target element. If it hit in the body, transfer is made to the label BODY.

Lines 41-44 determine whether the projectile hit the head. If so, the number of rounds to have hit the head is incremented and appropriate indices are set.

Lines 45-65 determine which body part of the target element is hit and the number of rounds that have hit that body part, and set the corresponding body part index L.

Lines 66-74 access the coefficients for the calculation of the probability of incapacitation given a hit in a specific body part.

Lines 75-76 calculate the probability of incapacitation for the target given that he is hit in a specific body part. The probability for the opposite mode of activity is also given.

Lines 77-97 store the body part probabilities of incapacitation for future calculation.

Lines 98-120 calculate the probability of total body

incapacitation and determine the expected time to that incapacitation. If CALC=0, the current round is not the last round in a burst and so the control returns to routine BRST.FIRE to see if the next round hits the target.

Lines 121-150 categorize the target into one of four sets according to his wounds and mode of activity.

Lines 151-196 determine the appropriate probabilities of different types of incapacitation which are then used in a call to Routine ATRIT to actually attrit the element.

APPENDIX E. LETHALITY, ACCURACY, AND CASUALTY INPUTS

Routine RES1 is used to reserve space for and to read in the lethality and accuracy arrays for Infantry small arms weapons. Additionally the arrays for casualty assessment play are reserved and read in by RES1. The specific arrays, their use and structure are given below.

AIMERROR

Contains the aim error measured in mils for small arms weapons. This is a two-dimensional array reserved as 5 by 7.

1st Dimension: Weapon type

1	M60/PKM
2	SAW
3	M16A1
4	AKM
5	M203

2d Dimension: Range to target

1	0-49 meters
2	50-99 meters
3	100-199 meters
4	200-299 meters
5	300-399 meters
6	400-499 meters
7	500-800 meters

BRST

Contains the mean burst size that a weapon type should fire indexed by the weapon type and ammunition type. This is a three-dimensional array reserved as 3 by 6 by 4.

1st Dimension: System type of firer

1	Tank
2	Mounted Infantry
3	Dismounted Infantry

2d Dimension: Weapon type of the firer

1	M16 Rifleman
2	M203 Grenadier
3	SAW/M60 Machinegun
4	RPG Antitank Gunner
5	AKM Rifleman
6	PKM Machinegunner

3rd Dimension: Ammunition type/mode of fire

1	AMMO1
2	AMMO2
3	AMMO3
4	AMMO4

DISPERSIONS Contains accuracy data for weapons capable of firing burst fire. This is a three-dimensional array reserved as 4 by 5 by 6.

1st Dimension: Type of mount

1	Tripod
2	Bipod
3	Prone
4	Standing

2d Dimension: Weapon type

1	M60 Machinegun
2	PKM
3	SAW Machinegun
4	M16A1/RPK/AK74
5	AKMS

3rd Dimension: Accuracy data

1	Horizontal distance from the first round to the center of impact of the subsequent rounds.
2	Vertical distance from the first round to the center of impact of subsequent rounds.
3	Deflection sigma of the distance from the first round to the center of impact of the subsequent rounds.
4	Elevation sigma of the distance from the first round to the center of impact of the subsequent rounds.
5	Deflection sigma of the dispersion of the subsequent rounds about their center of impact.
6	Elevation sigma of the dispersion of the subsequent rounds about their center of impact.

LAWHEAT Contains the lethality data for the LAW antitank weapon against moving and stationary targets. This is a six-dimensiona' array reserved as 2 BY 3 BY 7 BY 2 BY 6 BY 6.

1st Dimension: Target type

1	T62
2	BMP

2d Dimension: Speed of Target

1	Stationary
2	Moving less than 10 mph
3	Moving less than 20 mph

3rd Dimension: Range to target

1	0-49 meters
2	50-99 meters
3	100-149 meters
4	150-199 meters
5	200-249 meters
6	250-299 meters
7	300-350 meters

4th Dimension: Exposure of target

1	Fully exposed
2	Hull defilade

5th Dimension: Aspect Angle

1	0-30 degrees
2	31-60 degrees
3	61-90 degrees
4	91-120 degrees
5	121-150 degrees
6	151-180 degrees

6th Dimension: Outcome probabilities

1	KKILL
2	MKILL
3	FKILL
4	MFKILL
5	HIT
6	Expected Casualties

M203DISP

Contains the horizontal and vertical dispersions for the M203. This is a two-dimensional array reserved as 5 BY 2.

1st Dimension: Range

1	0-49 meters
2	50-99 meters
3	100-199 meters
4	200-299 meters
5	300-350 meters

2d Dimension: Dispersions
- 1 Horizontal
- 2 Vertical

M203DP — Contains the lethality data for the M203DP round against the BMP or BRDM. This is a four-dimensional array reserved as 2 BY 3 BY 7 BY 6.

1st Dimension: Target type
- 1 BMP
- 2 BRDM

2d Dimension: Target Disposure
- 1 Moving fully exposed
- 2 Stationary hull defilade
- 3 Stationary fully exposed

3rd Dimension: Range to the target
- 1 0-49 meters
- 2 50-99 meters
- 3 100-149 meters
- 4 150-199 meters
- 5 200-249 meters
- 6 250-299 meters
- 7 300-350 meters

4th Dimension: Lethality probabilities
- 1 MKILL
- 2 FKILL
- 3 MFKILL
- 4 KKILL
- 5 HIT
- 6 Expected casualties

VIPERHEAT — Contains the lethality data for the Viper weapons system. This is a four-dimensional array reserved as 3 BY 3 BY 5 BY 6.

1st Dimension: Target type
- 1 T72
- 2 BMP
- 3 BRDM

2d Dimension: Target Disposure
- 1 Moving fully exposed
- 2 Stationary hull defilade
- 3 Stationary fully exposed

3rd Dimension: Range to the target

1	0-99 meters
2	100-199 meters
3	200-299 meters
4	300-399 meters
5	400-500 meters

4th Dimension: Lethality probabilities

1	MKILL
2	FKILL
3	MFKILL
4	KKILL
5	HIT
6	Expected casualties

RPGHEAT

Contains the lethality data for the RPG against both moving and stationary targets. This is a four dimensional array reserved as 3 BY 3 BY 7 BY 6.

1st Dimension: Target type

1	XM1
2	IFV
3	ITV

2d Dimension: Target disposure

1	Moving fully exposed
2	Stationary hull defilade
3	Stationary fully exposed

3rd Dimension: Range to target

1	0-99 meters
2	100-199 meters
3	200-299 meters
4	300-399 meters
5	400-499 meters
6	500-599 meters
7	600-700 meters

4th Dimension: Lethality probabilities

1	MKILL
2	FKILL
3	MFKILL
4	KKILL
5	HIT
6	Expected casualties

A.CSLTY.DATA
B.CSLTY.DATA

Each of these arrays contains a coefficient of regression for casualty assessment play

C.CSLTY.DATA The first of these contains the coefficient "a", the second contains "b", and the third contains "c". Each is a four dimensional array reserved as 7 BY 2 BY 3 BY 5.

1st Dimension: Weapon type of projectile

1	M16/SAW (XM777 round)
2	M16A1 (M193 round)
3	M60/M14
4	M203 HE fragment
5	M18 Claymore
6	AKM/AK74
7	RPK/PKM

2d Dimension: Activity of target

1	Assault
2	Defense

3rd Dimension: Expected time to incapacitation

1	30 seconds
2	5 minutes
3	30 minutes

4th Dimension: Part of body hit by projectile

1	Head
2	Thorax
3	Abdomen
4	Pelvis and lower extremities
5	Upper extremities

APPENDIX F. DEFINITIONS

BDE.COMMANDER ATTRIBUTES

BDECUR	Represents the sum of BNWT attributes of each BN in the Brigade which has requested to move from the current coordination line.
BDELO	A lower bound. If BDECUR equals or exceeds BDELO each Battalion currently filed in the Brigade set to be given permission to move from the current coordination line.
BDEGO	The upper bound. If BDECUR equals or exceeds BDEGO, each Battalion currently in the Brigade set is ordered to move to their next coordination line.

BN.COMMANDER ATTRIBUTES

BNWT	Indicates the tactical weight or relative importance of a Battalion's position along a particular coordination line.
BNCUR	Represents the sum of the COWT attributes of each company in the battalion which has requested to move.
BNLO	The lower bound on BNCUR, which if equalled or exceeded by BNCUR causes each company in the battalion to have permission to move from their current coordination line.
BNGO	The upper bound on BNCUR which if equalled or exceeded by BNCUR constitutes an order for each company in the battalion to move to their next coordination line.
BATT	Indicates the number of the battalion.
BREQST	Indicates whether the battalion has requested to move from a particular phase line. 0 YES 1 NO
BNMSN	Indicates whether the battalion has permission to withdraw from a particular phase line. 0 NO 1 YES

COMPANY.COMMANDER ATTRIBUTES

P.TO.AREAS	A pointer to an array containing a sequence of position area numbers, called SEQUENCE.OF.AREAS.
COWT	Indicates the tactical weight or relative importance of the company's position along a particular coordination line.
CREQST	Indicates whether the company has requested to move from a particular phase line. 0 YES 1 NO
COMPY	Indicates the number of the company.
CMSN	Indicates whether the company has permission to withdraw from a particular phase line. 0 NO 1 YES

R4HAT	An integer used in calculating suppression which indicates the number of rounds received in the last minute as of three minutes ago.
R3HAT	An integer used in calculating suppression which indicates the number of rounds received in the last minute as of two minutes ago.
R2HAT	An integer used in calculating suppression which indicates the number of rounds received in the last minute as of one minute ago.
R1HAT	An integer used in calculating suppression which indicates the number of rounds received during the current minute.
R0HAT	Indicates the weights of all rounds suppressing the platoon during a thirty second portion of the battle.

R.MVR.CDR ATTRIBUTES

The entity R.MVR.CDR is used to count the attrition levels for user specified system and weapon types within the attacker maneuver unit.

R1DED R2DED R3DED R4DED R5DED R6DED	Indicates the number of dead systems in each of the six user defined system and weapon types belonging to R.MVR.CDR.
RUDED	Indicates the cumulative number of dead overall systems belonging to R.MVR.CDR.
RV.UNIT	Indicates the movement state of the unit 0 Stationary 1 Moving
R1SRT R2SRT R3SRT R4SRT R5SRT R6SRT	Indicates the initial number of systems alive at a designated start point for each of the six systems.
RUSRT	Indicates the total number of alive systems at a designated start point.
RETY	Indicates the appropriate plane of the three dimensional movement decision array that applies to R.MVR.CDR.
XCENT YCENT	Indicates the location of the centroid of elements in R.MVR.CDR.

B.MVR.CDR ATTRIBUTES

The entity B.MVR.CDR is used to count attrition levels for user specified system and weapon types within the defender maneuver unit.

BLUSIN	Indicates the number of defender force elements in B.MVR.CDR that are able to fire.
CUDED	Indicates the cumulative number of dead systems in each of the six user defined system and weapon types belonging to this entity.
CUSRT	Indicates the total number of alive systems at a designated start point.
MV.UNIT	Indicates the movement state of the unit. 0 Stationary 1 Moving
PRTY	Indicates the appropriate plane of the three dimensional movement decision array that applies to B.MVR.CDR.
REDRG	Indicates the range to the closest attacking force element who has fired at any element of B.MVR.CDR
REDSIN	Indicates the number of attacking (Red) force elements within a user specified range of B.MVR.CDR.
S1DED S2DED S3DED S4DED S5DED S6DED	Indicates the number of dead systems in each of the six user defined system and weapon types belonging to B.MVR.CDR.
S1SRT S2SRT S3SRT S4SRT S5SRT S6SRT	Indicates the initial number of systems alive at a designated start point for each of the six systems.

ATTRIBUTES OF TANK

ALIVE.DEAD — Indicates whether the entity is alive or dead.

0	Alive
1	Dead
2	Alive mounted in carrier.

AMMO1

TANK	IFV	ITV	BMP	DRAG	M16	M60/SAW	M203	RPG	AKMS	PKM	AIR	ADA
APDS	TOW	TOW	AGM	NA	LAW	LAW	LAW	AKM	NA	NA	AGM	SAM

AMMO2

TANK	IFV	ITV	BMP	DRAG	M16	M60/SAW	M203	RPG	AKMS	PKM	AIR	ADA
HEAT	NA	NA	NA	DRAG	VIP	VIPER	VIP	NA	NA	NA	RKT BMB	NA

AMMO3

TANK	IFV	ITV	BMP	DRAG	M16	M60/SAW	M203	RPG	AKMS	PKM	AIR	ADA
COAX	BM	NA	73 mm	M16 AR	M16	M60	M203 HE	RPG	AKMS	PKM	ATM WPN	NA

AMMO4

TANK	IFV	ITV	BMP	DRAG	M16	M60/SAW	M203	RPG	AKMS	PKM	AIR	ADA
NA	NA	NA	NA	NA	M16	SAW	M203 DP	NA	NA	NA	NA	NA

AMMO5

M16	M60/SAW	M203	RPG	AKMS	PKM	XM1
M18	M18	M18	NA	NA	NA	CAPDS

AMMO6

M16	M60/SAW	M203	RPG	AKMS	PKM	XM1
HG	HG	HG	HG	HG	HG	CHEAT

*NOTE: VIP=Viper, RKT=rocket, BMB= bomb, ATGM=AGM= antitank guided missile, DRAG=Dragon, BM=Bushmaster, M18=Claymore, HG=hand grenades, ATM WPN=automatic weapon.

AP.TOW	Same as AMMO1
AREA	Indicates the number of degrees the entity has in his initial search sector. Currently set to 90 degrees in BL.CREATE for ground elements and read in for air/ADA elements. It is adjusted in routine CHG.SEC.SEARCH. If there are no detections it is expanded to 360 degrees. If a target is detected, then the AREA is adjusted to 30 degrees by Routine TARGET.SELECT.
AREA.END	The number which identifies the ending movement area for the entity.
AREA.START	The number which identifies the starting movement area for the entity.
AW1.OR.MSL3	Same as AMMO3
AW2.OR.ADM	Same as AMMO4
BASE	Indicates the number of the base or airfield to which an aircraft is assigned.
BN	Indicates the number of the battalion to which the entity belongs.
C.1	Same as AMMO5
C.2	Same as AMMO6
CARR	Indicates the number of elements currently on board a personnel carrier which would normally dismount (e.g. the rifle squad members).
CBAR	Indicates the number of confusing forms in the background contrast of the vehicle.

CRF	Indicates the total number of rounds fired since the TANK last moved rounds into his ready rack.
CG.MUNITION	Designates whether or not AMMO1 for aircraft is a command guided munition. 0 NO 1 YES 2 Command guided with terminal homing.
CHECK.TIME	Indicator of the value of TIME.V when a specified event occurred. Used specifically to represent the time at which a TANK's next FIRE event is to occur.
CO	The number of the company to which the entity belongs.
COCDR	The element number of the entity's company commander. The company commander is the first entity input in his company.
COLOR	Indicates the color of the element. 0 Red (Attacker) 1 Blue (Defender)
CP.FINAL	The final control point that an aircraft encounters enroute to his objective.
CP.NEXT	The next control point that the aircraft will pass through.
CURR.SORTIE	The pointer of the sortie to which the aircraft belongs.
DEFNUM	The current position or activity of an element. 1 Full defilade 2 Turret defilade 3 Firing defilade 4 Stopping to fire (seeking concealment) 5 Moving (defilade determined by the terrain model) 6 Reached final area in movement.
DIR.OF.MVT	Indicates the entity's direction of

	movement measured in radians from East.
DIR.ON.RT	Indicates whether or not a vehicle is moving forwards or backwards on his route. 0 Vehicle is moving in order of increasing MCP numbers along the route. (forward) 1 Vehicle is moving in order of decreasing MCP numbers along the route. (backwards)
F.D	Indicates the accumulated percentage of of firepower damage sustained by the TANK.
FIP	Indicates whether or not the element is currently in the process of firing at a target. 0 NO 1 YES (fire in progress)
FIRED.AT	Indicates the total number of rounds fired at a TANK.
FIRING.MODE	Indicates the mode of fire that a dismounted element is to use.
FKILL	Indicates whether a TANK has sustained a firepower kill. 0 NO 1 YES
FOE	Indicates the pointer value of the TANK's current target.
FOM	Indicates whether or not an element is allowed to fire on the move. 0 NO 1 YES
FORM.CODE	The formation number of the formation to be used by the platoon. 0 Not in formation, vehicle moves along route without offset.
FORM.POS	The number indicating which place in the formation this element should occupy.
GRND.ATK	Indicates whether or not a vehicle is to accompany a ground attack. 0 NO, use regular SPD.LIMITS 1 YES, use DSMTD.SPDS

HE.DRAG	Same as AMMO2
HITSHOT	Indicates the number of hits (catastrophic kills) scored by a TANK during its current firing sequence.
HIT.STATE	An alpha variable indicating whether or not an element is alive or dead.
I.ROUTE	Indicates the route that an aircraft is currently using.
K.HIT	Indicates the number of hits sustained by a TANK which were sufficient to cause a catastrophic kill.
KKILL	Indicates whether the TANK has sustained a catastrophic kill. 0 NO 1 YES
LSS	Indicates whether an element has laser spot seekers. 0 NO 1 YES
LSD	Indicates whether an element has a laser designator/illuminator. 0 NO 1 YES
LSRON	Indicates whether the entity has his laser on in an operational mode.
LWR	Indicates whether an element has a laser warning receiver. 0 NO laser receivers 1 YES
LST.DIR	Indicates the last direction that the element was searching.
MFKILL	Indicates whether or not a TANK has sustained a simultaneous mobility and firepower kill. 0 NO 1 YES
MF.HIT	Indicates the number of hits sustained by a TANK which have caused simultaneous mobility and firepower damage.

MKILL	Indicates whether or not the TANK has sustained a mobility kill. 0 NO 1 YES
MICRO	Indicates the value of the micro terrain associated with the location the entity currently occupies.
MISSSHOT	Indicates the number of misses (anything other than catastrophic kill) scored by a TANK during its current firing sequence.
MODE	The attack mode which designates which helicopter tactics are to be employed. 0 Indirect rapid fire with another element guiding the round from another direct fire position 1 Ripple fire 2 Autonomous operations
MROLE	The mission role of an aircraft. 0 Aircraft at a base or forward attack area. 1 Aircraft enroute to an attack area 2 Aircraft returning to base from an attack 3 Aircraft employing pop-up tactics in an attack area 4 Aircraft at a control point for an attack area and moving to the attack area. 5 Aircraft employing a run in the dynamic attack tactics 6 Not currently used 7 All other aircraft (supply remotely piloted vehicles, etc)
MV.STATE	The primary control variable for initiating and stopping movement. 0 In position, do not move 1 First call to move, do a route select and start to move. 2 Continue movement along a previously selected route. 3 Stop along the route (e.g. stop to fire) 4 Next position has been

		reached, so stop
	5	Final position has been reached, never move again

M.BLUE.ALIVE — Indicates whether or not the element is a member of the set BLUE.ALIVE
- 0 Not a member
- 1 Member

M.COMP.UNIT — Indicates whether the element is a member of the company.
- 0 NO
- 1 YES

M.D — Indicates the accumulated percent of mobility damage sustained by a TANK.

M.HIT — Indicates the number of hits sustained by a TANK which have caused mobility damage.

M.PLT.UNIT — Indicates whether the element is a member of the platoon.
- 0 NO
- 1 YES

M.RED.ALIVE — Indicates whether the element is a member of the set RED.ALIVE.
- 0 Not a member
- 1 Member

M.TANKS — Indicates whether the element is a member of the set TANKS.
- 0 Not a member
- 1 Member

N.AB — Indicates the number of times a dismounted element has been hit in the abdomen.

N.ARM — Indicates the number of times a dismounted element has been hit in the upper extremities.

N.HD — Indicates the number of times a dismounted element has been hit in the head.

N.PLE — Indicates the number of times a dismounted element has been hit in the pelvis or lower extremities.

N.TH — Indicates the number of times a

	dismounted element has been hit in the thorax.
NAME	The element number of the entity
NEXT.MCP	Movement control point number(on the designated route) toward which the element is now moving 0 End of the route has been reached
ND.HIT	Indicates the number of hits sustained by a TANK which caused no damage
NO.OBS	The number of visual observers that are actively engaged in the target acquisition process. For an aircraft with 3 crew members, this could be 1,2, or 3 depending on the crew duties.
NUM.HIT	Indicates the total number of hits sustained by a TANK. This includes no damage hits.
P.AB.INCAP	Indicates the probability that an element will become incapacitated given that he is hit in the abdomen.
P.ARM.INCAP	Indicates the probability that an element will become incapacitated given that he is hit in the upper extremities.
P.BODY.INCAP	Indicates the probability that an element suffers some degree of incapacitation. This is the total body probability of incapacitation and considers all rounds that have impacted on the target.
P.HD.INCAP	Indicates the probability that an element will become incapacitated given that he is hit in the head.
P.PLE.INCAP	Indicates the probability that an element will become incapacitated given that he is hit in the pelvis or lower extremities.
P.TH.INCAP	Indicates the probability that an element will become incapacitated given that he is hit in the thorax.

PH	Indicates the sensing of the last round 1 First round 2 Hit 3 Unsensed miss 4 Sensed miss
PI.HAT	Indicates the terrain complexity factor of the terrain in which the entity is located.
PLOW.COND	Indicates the availability of a mineplow on a tank. 0 No plow available 1 Plow available but not in use 2 Plow being used
PLT	The number of the platoon to which the entity belongs.
PLTLDR	The element number of the entity's platoon leader. The platoon leader must be the first element input in his platoon.
POP.UP	Indicates whether or not pop-up tactics are to be used 0 Dynamic tactics 1 Pop-up tactics
POP.UP.TIME	For dismounted elements the pop-up time is the time at which he will no longer be suppressed by artillery that has impacted. For aircraft utilizing pop-up tactics it is the time at which the aircraft last unmasked.
POS.IN.PLT.AREA	Indicates the position in the platoon area for an element. In the ground model this number is used to access the POSITION array for the (x,y) coordinates for an element within the platoon. The air-air ground model uses this for dispersing aircraft at bases.
POS.POINT	Indicates the location that the entity uses to access the SEQUENCE. OF.AREAS array
PRI.DIR	The entity's primary direction of search.
PROJO	The number of the ammo type which the entity has selected to fire (e.g. if

the element uses AMMO3 then PROJO(TANK)=3)

RADAR	Indicates whether an element has an acquisition radar 0 NO 1 YES
RANGE	Indicates the range from the entity to his current target in meters at the time of fire.
R.CON	Indicates the reload condition of a tank. Used only for XM1s. Indicates the status of ammunition movement into the ready rack.
RDRON	Indicates whether or not the radar belonging to an element is on or off. 0 Off 1 On
R.D.STATUS	Indicates whether an entity is attempting to reach his foxhole after a dismount or whether he is attempting to reach his vehicle after being ordered to remount. 0 Remount, go to vehicle 1 Dismount, go to foxhole
RGT	Indicates the number of the regiment to which the element belongs (if applicable)
ROUTE	Indicates the number of the route along which the element is travelling 0 Not using a route
RWR	Indicates whether a system has radar warning receivers 0 NO 1 YES
SCHED	Indicates the number of FIRE events scheduled. 0 NONE
SCH.TO.MOVE	Indicates whether Event MAX.WAIT.TIME has been scheduled for this entity 0 NO 1 YES
SEC	Indicates the number of the section to which the entity belongs

SECLDR	The element number of the entity's section leader. The section leader must be the first element in his section.
SPD	The entity's speed at the end of the most recent movement update.
SQDVEH	Indicates the number of the entity's squad vehicle
SYS.TYPE	This represents the general class of the system of the entity 1 TANKS 2 Mounted Infantry 3 Dismounted Infantry 4 Artillery 5 Air 6 Air Defense 7 Supply 8 Comm/EW/ACQ/Intel 9 Other
THERMAL	Designates units with thermal imaging equipment 0 No thermal equipment 1 Thermal equipment
TRF	Indicates the total number of rounds that an entity has fired
T.SPD	The simulation time at which the most recent movement update ended. (The time that SPD was last set)
VAREA	The angular number of degrees that aircraft and air defense (except long range ADA without visual observers) observe. This is not to exceed 360 degrees.
VEH.TYPE	The vehicle type for a given weapon system. This indicates a particular lethality configuration of a weapon system.
WPN.TYPE.	Describes the specific system within the system code 1 XM1 (105mm) 2 XM1 (120mm) 3 IFV 4 ITV 5 DIVAD

6	DRAGON
7	T72 Tank
8	BMP
9	ZSU 23-4
10	M16 Rifleman
11	M60/SAW Machinegunner
12	M203 Grenadier
13	RPG Antitank Gunner
14	AKMS Rifleman
15	PKM Machinegunner

X.CURRENT	The X-coordinate for the entity as of the last movement update
Y.CURRENT	The Y-coordinate for the entity as of the last movement update
Z.CURRENT	The elevation for the entity as of the most current movement update

A.CSLTY.DATA	Contains the A coefficients for casualty assessment equations.
AA.LS	Variable used to calculate the intersection of the line of sight with a forest ellipse.
AB.IN	Indicates the probability of incapacitation for the opposite mode of activity than the target is currently in for the case in which he is hit in the abdomen.
AB.TH.WIDTH	Indicates the width of the abdomen and thorax area of a soldier
ABORTPRINT	Indicates whether the user desires the number and the reasons for aborted fire events to be printed at the end of the simulation 0 NO 1 YES
AC.DELTA.T	Indicates the time interval between successive target acquisition events for aircraft detection of ground elements or air elements and for all ground elements detecting aircraft.
ABDO.HT	Indicates the height of the abdomen area of a soldier.
AC.RDCHECK.TIME	Indicates the time in seconds between successive redeployment checks for all redeployable aircraft.
ACCBM	A data array containing accuracy data for the BMP 73mm fired at stationary targets.
ACCHT	A data array containing accuracy data for tank HEAT rounds fired at stationary targets.
ACCKE	A data array containing kenetic energy round accuracy data for rounds fired at stationary targets.
ACCMSL	A data array containing accuracy data for missile rounds fired at stationary and moving targets.

SETS IN STAR

AC.SCHEDULE	A set which contains the temporary entities AIR.SORTIE that have a mission scheduled. This set is owned by the system.
BATTALION	A set which may contain the permanent entity COMPANY.COMMANDER. This set is owned by the BN.COMMANDER.
BLUE.ALIVE	A set containing all Blue elements that are alive in the simulation. This set is owned by the system.
BRIGADE	A set which may contain the permanent entities BN.COMMANDER. This set is owned by the permanent entity BDE.COMMANDER.
COMP.UNIT	A set which may contain the temporary entity UNIT (TANK) which indicates that the entity belongs to the company. This set is owned by the permanent entity COMPANY.COMMANDER.
ON.THE.GROUND	A set which may contain the temporary entities AIR.SORTIE. This set indicates that the AIR.SORTIE is on the ground and not scheduled for a mission. The system owns this set.
RED.ALIVE	A set containing all Red elements that are alive in the simulation.
STOCKS	A set which may contain the temporary entity AMMO.PILE. This set is owned by the permanent entity SUPPLY.OFFICER.
TANKS	A set containing all UNITS(TANKs) currently alive in the simulation. This set is owned by the system.
THE.SKY	A set containing the temporary entities AIR.SORTIE. Membership indicates that the AIR.SORTIE is currently flying. This set is owned by the system.
WANT.AIR	A set which may contain the temporary entities BLU.AIREQST.

ADDON	A data array containing accuracy "add-ons" to account for the dispersion caused by a firer who is moving.
AIR.SPEED	Indicates the average cruising speed for horizontal flight, the average speed (climbing) for vertical or near vertical flight, the estimated minimum speed for flight at or near zero altitude, and the minimum altitude at which the horizontal speed can be maintained.
ARG.LS	Used to calculate the intersection of a line of sight with a forest ellipse
ARM.IN	Used to calculate the probability of incapacitation to dismounted elements.
ATK.POS	Contains the attack area position information for STAR-AIR
B.BDE.ROUTE	Indicates the last Blue preplanned route, which is the route that crosses the Blue defensive area connecting adjacent battalion assembly areas. Used in STAR-AIR
B.CSLTY.DATA	Contains the B coefficients for the casualty assessment equations.
B.NUM.ALIVE	Indicates the total number of Blue elements to be represented in the simulation.
B.PCT.ATT	Indicates the Blue force percent attrition which is used to stop the simulation. If the Blue forces reach or exceed this value, the simulation will halt.
BA	Indicates the number of APDS rounds stored in the hull compartment.
BASE.LS	Indicates the minimum elevation of the terrain above sea level.
BB.LS	Used to calculate the intersection of the line of sight with a forest ellipse
BBBPOINT	Contains the pointer values of all Blue TANKs
BBN	Indicates the total number of Blue Battalion sets to be created
BC.COUNT	Indicates the number of Red TANKs created.

	This is used as a reference value by the Blue TANKs.
BDI	Used to tally the number of times a Blue element enters Event FIRE.
BH	Indicates the number of HEAT rounds stored in the hull compartment.
BLCOMP	Indicates the number of defender (Blue) maneuver commanders in the simulation.
BLIST	Indicates the number of elements on an element's LIST upon a call to TARGET.SELECT. Used to accumulate statistics.
BLUE	Indicates the color of the defending force
BLUGUYS	Indicates the number of defender elements to be created during the current creation of Blue forces.
BLUTNUM	Indicates the total number of times that Blue forces are to be created.
BM.MOV	A data array containing accuracy data for the BMP 73mm against moving targets.
BSTD	Indicates the value of BDET.TIME in Routine STEP.TIME. Used to tally the statistics for Blue STEP.TIME detections.
BTIME	Indicates the total time to move rounds from the hull compartment to the ready bustle measured in seconds.
BWD.LOOK	Used in conjunction with FWD.LOOK in determining which lines of sight should be calculated. If BWD.LOOK=1 , line of sight should be calculated from element B to element A. If FWD.LOOK=1, line of sight is to be calculated from element A to element B. If the value of both variables is equal to 1, line of sight is to be calculated between both elements.
C.BAR	Indicates the number of confusing background figures in the detection routines.
C.CSLTY.DATA	Contains the C coefficients for the casualty assessment equations

CAPDS	Indicates the number of APDS rounds initially loaded in the bustle ready rack.
CASE	Used by Routine RELOAD to take appropriate ammunition redistribution action. 1 40 rds XM1 120mm 2 43 rds 3 49 rds 4 52 rds or 55 rds XM 105mm
CASEAP	Indicates the number of APDS rounds initially loaded on the XM1.
CASEHE	Indicates the number of HEAT rounds initially loaded on the XM1.
CC.LS	Used to calculate the intersection of the line of sight with a forest ellipse.
CDA	Indicates the number of APDS rounds loaded next to the left front fuel tank and in the swing basket.
CDH	Indicates the number of HEAT rounds loaded next to the left front fuel tank and in the swing basket.
CDTIME	Indicates the total time to move rounds from the fuel compartment/basket to the ready bustle in seconds.
CHEAT	Indicates the number of HEAT rounds initially loaded in the bustle ready compartment.
CHTMAX.LS	Indicates the tallest forest feature intersected by the line of sight.
CNUM	Indicates the total number of company sets to be created. (i.e COMPANY.COMMANDER entities)
COM.VAL	A value between 0 and 1 which represenets the probability of communicating with platoon members during execution of the platoon fire coordination logic.
CPK.LS	Indicates the height of a given forest.
CRIT.H	A computing elevation of terrain array parameter.
CRITICAL.VALUE	A value between 0 and 1 which is compared to the percent of the target visible to the observer.

Line of sight does not exist if the percent visible is less than CRITICAL.VALUE.

CSET — An indicator used in Routine BL.CREATE to indicate whether or not Routine COORD.SET has been called.

CS1.LS
CS2.LS — Arrays containing the intersections of Line of Sight with forest ellipses.

CVHTV.LS
CVHTW.LS — Indicates the forest height at V or W.

CYCLE.TOT — Trouble shooting aid for use in STAR-AIR.

DAM.ARRAY — Contains the possible values for HIT.STATE

1	NDAM
2	MDAM
3	FDAM
4	MFDM
5	DEAD
6	MISS

DAMAGE.NUM — Indicates the damage status of an entity after having a round impact on or near it.

1	Hit but already MF killed
2	Mobility damage
3	Firepower damage
4	Mobility and firepower damage
5	Catastrophic kill
6	Miss

DELTA.T — Represents the time interval in seconds between successive schedulings of event STEP.TIME.

DFCNV.LS — Used in Routine Newton to assist in calculating the lowest line of sight line over a given hill. This is the derivative of FCNV.LS.

DRAGON — Indicates the number of Dragon rounds initially allocated to each Dragon missile team.

DUM.I — Used to simplify coordinate computation in accessing LIST.H.

ELV.LS — Used in Routine Newton as a computational variable.

EQ.LS — Used to simplify computations in Routine LOS.

EXPOSED.LIMIT Indicates the amount of time a pop-up aircraft

will remain exposed before masking or going into defilade.

FCNV.LS — Used in Routine Newton to assist in the calculation of the lowest line of sight line over a given hill.

FF1 — These variables are used to count the number of times firing events are aborted due to various reasons See F1-F9.

FF2

FF3

FF4

FF5

FF6

FF7

FF8

FF9

FOR.CHG.INT — Indicates the distance within which formation changes are accomplished.

FORM.OFFSET — Contains the x,y offsets that the entity has for his position in the platoon formation. Used primarily for vehicles.

FQ.LS — Used to simplify computations in Routine LOS.

FRRDRON — Indicates the fraction of time that a given radar system is allowed to operate. If blinking of the radar is not desired, the input value should be 1.

FSQ.LS — Indicates the square of FQ.LS.

FWD.LOOK — Used in conjunction with BWD.LOOK. If FWD.LOOK=1, line of sight is calculated from element A to element B. If BWD.LOOK=1, line of sight is to be calculated from element B to element A. If the value of both variables is equal to one, line of

	sight is to be calculated in both directions.
F1	The variables F1 to F9 are used to tally the reasons for aborted fires in F.PRINT. F1= FOE of the firer does not equal ID.
F2	CHECK.TIME does not equal TIME.V
F3	FIP=1 (i.e. There is already a fire in progress)
F4	The firer is already dead or firepower killed.
F5	The firer is in full defilade.
F6	The selected target is in full defilade.
F7	The percent visible is less than the CRITICAL.VALUE.
F8	The target is dead.
F9	The target is now out of range.
G.AMM	Indicates the value of GAMMA in Routine GEOM measured in radians. It is printed out in the shot list.
GQ.LS	Used to simplify line of sight computations.
GSIZE	Indicates the size of a grid square in meters.
H.IN	Used to calculate the probability of incapacitation for dismounted elements.
HCC	A flag variable used in Event STEP.TIME to indicate whether or not the Blue forces have commenced rearward movement.
HHV.LS	Terrain elevation where LOS is tangent to a hill.
HHW.LS	Indicates the height of a hilltop being checked to see if it interrupts LOS.
HIDE.TIME	Indicates the amount of time that a pop-up aircraft will remain in defilade when it hides.
HT.E	An array containing the height of all the forest ellipses.
HT.H	An array indicating the maximum height of the

	"normal" underlying curve describing this hill mass.
HT.LS	Indicates HT.H for a single hill.
HT.MOV	An array containing accuracy data for tank HEAT rounds firing against moving targets.
HTS.LS	A computational variable used to indicate the cover height at S.
HTV.LS	A computational variable used in Routine NEWTON.
IEL.LS	An array containing the indices of the forest ellipse intersections with line of sight.
IGX.LS	An array containing the X indices of the grid squares that the line of sight crosses.
IGY.LS	An array containing the Y indices of the grid squares that the line of sight crosses.
INF.CAS.ASS	Indicates whether the user desires casualty assessment to be simulated in detail for the ground elements. This refers to calculating the body part hit by the projectile etc. The default is that the probability of kill given hit equals one. 0 Simple casualty assessment 1 Detailed casualty assessment
IN.ECHO	Indicates whether the user wishes to echo print all of the input data after it has been read in. 0 NO 1 YES
INFANTRY.POSITION	Contains the position data for the dismounted elements in a given platoon
ISGX.LS ISGY.LS	Temporary variable used to assist in computing IGX and IGY in Routine LOS.
IX.LS IY.LS	Indicates the X and Y indices of a particular grid square. Used by Routine LOS.
KCREP	A counter used in Routine LOS to count cover ellipses.
KE.MOV	An array which contains accuracy data

for kenetic energy rounds fired at moving targets.

KHREP	A counter used in Routine LOS to count hills.
KILLED.MOUNTED	Indicates whether the entity in question was killed while mounted on his vehicle or while dismounted. This is used as a flag to conduct estimated casualty assessment against the soldiers in the vehicle.
KILLER	A variable used to tally the number of calls to Routine F.PRINT.
KTREP	A counter used to avoid processing any ellipse or hill more than once during any line of sight call. It is incremented by 1 for each call to LOS. The values of KCREP and KHREP are compared to KTREP.
LAG.LS	Indicates the sum of LAGA.LS+LAGB.LS. 0 A and B are both ground elements. 1 Either A or B is a ground element, the other is an air element. 2 Both A and B are air elements.
LAGA.LS	Indicates whether element A is a ground or air element. 0 Ground 1 Air
LAGB.LS	Indicates whether element B is a ground or air element. 0 Ground 1 Air
LATOB.LS	Indicates whether a line of sight should be calculated from element A to element B. 0 YES 1 NO
LBTOA.LS	Indicates whether a line of sight should be calculated from element B to element A. 0 YES 1 NO
LIM.SPDS	Indicates the limiting speeds for vehicles

and ground dismounted elements during movement.

LIN — Indicates the total number of calls made to the line of sight routine Routine LOS. This is tallied as the number of PC.VIS.

LINE.OF. SIGHT.EXISTS — Indicates whether PC.VIS is greater than CRITICAL.VALUE.

0	NO
1	YES

LIST — Contains the pointer values of a TANK's detected targets. Each TANK has a list which is accessed by referencing the pointer value of the LIST.

LIST.C — Indicates the number of forest ellipses overlapping a selected grid square. This array also contains the ellipse numbers if there are any.

LIST.H — A list of hill numbers for each grid; also a BASE value.

LOAD.UP — Indicates the estimated time in seconds to place pallets of ammunition on the supply vehicle at the ATP.

MAX.REMOUNT. WAIT.TIME — Indicates the maximum time that a vehicle or unit of vehicles is to wait after a move call before departing for his subsequent position. This waiting time then reflects how much time his riders have to move from their foxholes to the vehicle before it will depart without them.

MAXCARR — Indicates the maximum number of elements a vehicle could mount both in and on the vehicle. This is used as an upper bound on the total number of riders a vehicle will be able to mount from other disabled vehicles in his platoon before it has reached full capacity.

MNDRNG — Indicates the range at which any element is presumed to automatically detect any opposing element to whom line of sight exists. Used in TAR-ACQ

MNDTIME	Indicates the detection time for all detections resulting from automatic detections for minimum range (MNDRG) Used in STAR-AIR
MOV.BLU	Indicates whether the Blue forces are to be created stationary or moving. 0 Stationary 1 Moving
MOVE.DATA	Contains area and route numbers for the platoons.
MU	Contains a parameter of the Normal distribution associated with the AMSAA supplied log-normally distributed load times for various vehicle types as a function of range to the target.
MXBLIST	Tally variable indicating the maximum of BLIST.
MXRLIST	Tally variable indicating the maximum of RLIST.
N.CAS	The number of rows to be created in the counterattack array CA.DATA. 0 No counterattack data is to be read
N.CORAIR	Indicates the number of Corps Air augmentations available in the air model.
N.CURR.ECH	Indicates the current echelon number of the last echelon created.
N.ECHELONS	Indicates the total number of attacker echelons (Red) to be created during the battle.
NBID	A variable tallied as the number of BDI.
NBLIST	Tally variable which indicates the number of list size checks on BLIST.
NBSTD	Indicates the total number of Blue element STEP.TIME detections
NCASE	Indicates the number of this replications of the simulation.

	This must be an integer from 0-999.
NCOS	Indicates the total number of company sized elements created in the simulation.
NCT.LS	Indicates the number of Newton iterations performed in Routine Newton.
NCVELS	Indicates the number of forest ellipses on the battlefield.
NELS.LS	Indicates the number of forest ellipses along this line of sight.
NGRIDX	Indicates the number of grid squares for the battlefield terrain in the X direction
NGRIDY	Indicates the number of grid squares for the battlefield in the Y direction.
NGRSQ.LS	Indicates the number of grid squares along this line of sight.
NHILLS	Indicates the total number of hills represented in the terrain model. (i.e. hills on the battlefield)
NILO NIUP	Indicates the time to detect a firing target given that the observer is looking in a 90 degree sector containing the target at the time of fire. This time is a Uniform number on the interval (NILO, NIUP) (i.e. ninety lower, ninety upper)
NO	Used for logical "IF" checks. Set equal to 0.
NO.ATK.AREA	Indicates the total number of both Blue and Red preplanned attack areas, to include assembly areas and bases. Used in STAR-AIR
NO.POS	Indicates the number of preplanned attack positions within an attack area.
NO.ROUTES	Indicates the total number of permanent air routes to be input.
NO.SORTIES	Indicates the total number of AIR.SORTIES to be input, excluding any air augmentation from Corps.
NRBNS	Indicates the total number of Red battalions input into the simulation.

NRID	Tallied as the number of RDI.
NRLIST	Tally variable indicating the number of list size checks on RLIST.
NRSTD	Indicates the number of Red STEP.TIME detections.
NUM.PILES	Indicates the number of AMMO.PILE temporary entities are to be created. Note that an ATP is not an AMMO.PILE.
NUMBER.OF. SYSTEMS	Used in MAIN to dimension the SYS.TYPE by WPN.TYPE arrays.
N1WPNS	Indicates the total number of weapon systems with PROLE=1 (i.e. Air Defense). Used in STAR-AIR
N2WPNS	Indicates the total number of systems with PROLE=2 (i.e. Aircraft).
OFFLOAD	Indicates the estimated time in seconds to place pallets of ammunition on the ground at the AMMO.PILE.
OFFSET	Contains the x,y offsets that elements are to maintain from their route element leader during movement.
ONDISK	Indicates whether or not the user desires the shot list and final attribute list to user specified disk files. 0 NO, paper output only 1 YES
P.V	Indicates the value of PC.VIS for a line of sight call.
PCA.UNC	Indicates the percent of element A which is uncovered. This is determined by Routine LOS.
PCA.VIS	Indicates the percent of the element that is visible to the element trying to detect him. This is determined by Routine LOS.
PCB.UNC	Indicates the percent of element B which is uncovered. This is determined by Routine LOS.

PCB.VIS	Indicates the percent of element B which is visible to an entity trying to detect him. This is determined by Routine LOS.
PCF1	Indicates the pallet conversion factor for XM1 APDS 120mm rounds. This converts pallets of ammunition to actual rounds. PCF1=30 rounds
PCF2	Analogue of PCF1 for XM1 HEAT rounds. PCF2=30 rounds
PCF3	Analogue of PCF1 for TOW rounds. PCF3=12 rounds
PCF4	Analogue of PCF1 for DRAGON rounds. PCF4=20 rounds
PCF5	Analogue of PCF1 for 25mm Bushmaster. PCF5=500 rounds
PCF6	Analogue of PCF1 for .50 caliber machinegun. PCF6=1500 rounds
PC.VIS	Indicates the percent visible of a target at impact as scaled by TARDIM(SYS.TYPE, WPN.TYPE,11).
PEAK.H	Indicates the height of a hilltop measured from sea level=0.
PHAROW	Indicates the currently occupied coordination line (i.e. currently occupied by the Brigade) This is used in the movement coordination logic.
PHZLINES	Indicates the total number of coordination lines for defending units in the simulation. This is used in the movement coordination logic.
PK.LS	Indicates the peak value for this hill.
PLE.IN	Used to calculate the probability of incapacitation for dismounted elements.
PLE.WIDTH	Indicates the width of the pelvis and lower extremity area of a soldier.
PLACES	Indicates the movement areas, pointer variables to AMMO.PILES, the ground force area associated with the AMMO.PILE, and the

	predetermined pallet loading loading plan for supply vehicles servicing a particular AMMO.PILE.
PNUM	Indicates the total number of platoon sets to be created. (i.e. PLATOON.LEADER entities)
POINT.HOLD	Contains tactics and firing information for each weapon type.

1 Attempt to engage your platoon leader's target; failing this, search your LIST for the highest priority target not already engaged; use your alternate ammunitition type.

2 Attempt to engage your platoon leader's target; failing this, search your LIST for the highest priority target not already engaged; use the specified ammunition type from the TARGET.SELECTION array.

3 Attempt to engage your platoon leader's target; failing this, search your platoon for the highest priority unengaged target.

4 Same as 1 but the company is searched

5 Same as 2 but the company is searched

6 Same as 3 but the company is searched

7 Attempt to acquire your platoon leader's target; failing this, engage your highest priority target

8-14 Same as 1-7 except do not attempt to acquire your platoon leader's target

POS.PLOT	Indicates the time interval in which position plots are to be made. This is used in conjunction with XYCARDS.
POSITION	Contains the element x,y coordinates of positions indexed by the platoon and area number.
POW.LS	A computational variable used in Routine LOS.
PVG PVH PVM	Defines the level (percent) of visual vertical search effort expressed as a fraction expended for ground level, high level, and medium level search respectively. The sum of PVG + PVH + PVM = 1. This is used in STAR-AIR
PXX.E	A variable used to compute the elevation

of terrain ellipses in Routine LOS.

PXX.H	A variable used to compute the elevation of terrain ellipses in Routine LOS. used in Routine LOS.
PXX.LS	A variable used to compute the elevation of terrain ellipses in Routine LOS.
PXY.E	A variable used to compute the elevation of terrain ellipses in Routine LOS.
PXY.H	A variable used to compute the elevation of terrain ellipses in Routine LOS.
PXY.LS	A computing variable used in Routine LOS.
PYY.E	A computing variable dealing with ellipses in Routine LOS.
PYY.H	A computing elevation of terrain parameter used in Routine LOS.
PYY.LS	A computing variable used in Routine LOS.
R.MED R.MIN	The vertical angle measured from horizontal separating the ground level to medium level and the medium level to high level (altitude) search effort. Since a discrete visual search includes a 30 degree angle, R.MIN and R.MED should be 30 degrees apart. Both are input in degrees and converted to radians. Used in STA -AIR.
R.NUM.ALIVE	Indicates the total number of Red elements to be represented in the simulation.
R.PCT.ATT	Indicates the percent attrition for RED which is used to stop the simulation. If the RED forces reach or exceed this value, the simulation will halt.
RBN	Indicates the number of attacker battalion commanders.
RC.COUNT	Indicates the number of BLUE TANKs created. This is used as a reference value by the RED TANKs.
RDI	A variable used to tally the number of times that a RED element enters Event

FIRE.

RED	Indicates the color of the attacking forces.
RLIST	Indicates the number of targets on a RED element's LIST. Used to accumulate statistics on the number, maximum and mean LIST size.
ROUTE.DATA	Contains the coordinates for each MCP and formation code to be used by the element's platoon.
RRRPOINT	Contains the pointer values of all of the RED TANKs.
RSTD	Indicates the value of RDET.TIME upon a return to Event STEP.TIME. It is used to tally detection statistics.
RX.LS RY.LS	Computational variables used in Routine LOS.
SIM.STOP	Indicates the latest time at which the simulation is to stop. This is measured in simulated seconds.
SIZEA.LS	Indicates the vertical height dimension of element A. The percent visible is taken as a fraction of this size.
SIZEB.LS	Indicates the vertical height dimension of element B. The percent visible is taken as a fraction of this size.
SQ.LS	A computing variable used in Routine LOS.
SS.LS	Indicates the forest boundary used in checking to see if the forest edge interrupts line of light.
STEP.AIR	Indicates how closely a dynamically generated flight path will trace the contour of the terrain over which the aircraft is flying. Used in STAR-AIR
S1.LS	Indicates the intersection of line of sight with a forest ellipse in line of sight computations.

S1A Indicates the number of APDS rounds loaded in the bustle semi-ready compartment.

S1H Indicates the number of HEAT rounds loaded in the bustle semi-ready compartment.

S1TIME Indicates the total time to move first increment of rounds from the semi-ready to the ready bustle measured in seconds.

S2.LS Indicates the intersection of line of sight with a forest ellipse in LOS computations.

S2A Indicates the number of APDS rounds loaded in the bustle semi-ready compartment.

S2H Indicates the number of HEAT rounds loaded in the bustle semi-ready compartment.

S2TIME Indicates the total time necessary to move the second increment of rounds from the semi-ready to the ready bustle measured in seconds.

T.BLUGUYS Indicates the simulation time at which the next creation of defender (BLUE) elements is to occur.

T.CURR.ECH Indicates the time at which the current echelon is created.

TARDIM Contains the target dimensions of all entity types in the simulation. This is indexed by the SYS.TYPE and WPN.TYPE of the element.

TARGET Contains the pointer values to a TANK's detected target list (LIST) and the pointer value to the TANK.

TEMP.TGT Used to temporarily store the pointer values from a TANK's list of detected targets. (LIST)

TEMPL A temporary holding array used in MAIN to hold values which will eventually be stored in TABLE and RTAB.

TH.IN Indicates the probability of incapacitation for the opposite mode of activity than the target is currently in for the case in which

he is hit in the thorax.

THLO THUP	Indicates the time to detect a firing target if the observer is looking in a thirty degree sector containing the firing element at the time of fire. This time is Uniform (THLO,THUP). (i.e. thirty lower, thirty upper)
TIM.IN	Indicates the amount of time that is to pass before an element is incapacitated.
TMACA.LS	Indicates the macro-terrain elevation for element A computed by Routine ELEV for line of sight determination.
TMACB.LS	Indicates the macro-terrain elevation for element B computed by Routine ELEV for use in line of sight determination.
TMICA.LS	Indicates the micro-terrain offset (+/-) from the macro-terrain associated with element A's position.
TMICB.LS	Indicates the micro-terrain offset (+/-) from the macro-terrain associated with element B's position.
TOO.CLOSE	Indicates the distance in which proximity detections occur. If a tank is within TOO.CLOSE meters of a detected tank at the time of detection, both tanks are detected.
TOW1CASE	Indicates the number of TOW rounds initially allocated to Infantry Fighting Vehicles or Cavalry Fighting Vehicles.
TOW2CASE	Indicates the number of TOW rounds initially allocated to Improved TOW Vehicles.
TRACE.TIME	Indicates the time at which diagnostics are turned on.
TTIME	Used to activate a debugging traceback feature built into numerous STAR Events and Routines. If activation is not desired, make TTIME greater than SIM.STOP.

TTT	Indicates the total number of rounds fired.
TWOGV.LS	A computational variable used in Routine NEWTON.
TWOXBA.LS	A computational variable used in Routine LOS.
TWOYBA.LS	A computational variable used in Routine LOS.
VEH.TACTIC	Indicates which elements if any are to overwatch the dismount and subsequent attack (if applicable) of dismounting elements. 1 Tanks overwatch 2 BMPs overwatch 3 Tanks and BMPs overwatch 4 Neither overwatch
V.LS	Indicates the coordinate tangency of of a line of sight with a hill.
VISFRA.LS	Indicates the visible fraction of element A (i.e. fraction of SIZEA) which can be seen by element B.
VISFRB.LS	Indicates the visible fraction of element B (i.e. fraction of SIZEB) which can be seen by element A.
VM.LS	A computational variable used in Routine NEWTON.
VSUB.LS	A computational variable used in Routine NEWTON.
W.L	Indicates the location of a hilltop being checked to see if it blocks line of sight in Routine LOS.
WH.1 WH.2 WH.3 WH.4 WH.5 WM.1 WM.2 WM.3	Indicates a user input for the number of rounds that may be fired prior to activating a user specified tactic. Variables preceded by WH are used by Routine WE.HIT immediately following a catastrophic kill from a firing defender element and those preceded by WM are used by Routine WE.MISS immediately following any other result
X.LO.BDRY	Indicates the X coordinate of the

southwest corner of the battlefield in meters.

XA.LS	Indicates the X coordinate of element A on the battlefield. Used in Routine LOS to calculate the line of sight from element A to element B.
XB.LS	Indicates the X coordinate of element B on the battlefield. Used in Routine LOS to calculate the line of sight from element B to element A.
XBA.LS	Used for computations to establish the grid square list for Routine LOS.
XBASQ.LS	A computational variable used in routine LOS indicating the square of XBA.LS.
XC.E	Indicates the X coordinate of an ellipse center for a given ellipse of a given hill.
XC.H	Indicates the X map coordinate of the center of a given hill.
XINC.LS	Used to compute grid squares crossed by a line of sight in Routine LOS.
XS.LS	A computational variable used in Routine LOS.
XSTEP.LS	Used for computations to establish the grid square list in Routine LOS.
XYBA.LS	A computational variable used in Routine LOS.
XYCARDS	Indicates whether the user desires the (x,y) coordinates, NAME, ALIVE.DEAD status, SYS.TYPE, and WPN.TYPE of each element in the battle to be output every POS.PLOT seconds. 0 NO 1 YES
Y.LO.BDRY	Indicates the Y coordinates of the southwest corner of the battlefield. This is stored in meters.
YA.LS	Indicates the Y coordinate of element A on the battlefield. This is used in line of sight computations in Routine LOS.

YB.LS	Indicates the Y coordinates of element B on the battlefield. This is used in line of sight computations in Routine LOS.
YBA.LS	Used for computations to establish the grid square list in Routine LOS.
YBASQ.LS	A computational variable used in Routine LOS to indicate the value of the square of YBA.
YC.E	Indicates the Y coordinate of the center of a given ellipse of a given hill.
YC.H	Indicates the map coordinates of the center location for a hill.
YES	Indicates a value of 1. Used in conditional statements.
YINC.LS	A computational variable used in computing grid squares crossed by a line of sight in Routine LOS.
YS.LS	A computational variable used in routine LOS.
YSTEP.LS	Used for computations to establish the grid square list in Routine LOS.
ZA.LS	A computational variable dealing with elevation of a given element A in Routine LOS.
ZB.LS	A computational variable dealing with the elevation of element B in Routine LOS.
ZBA.LS	A computational variable used in Routine LOS to indicate ZB.LS-ZA.LS.
ZH	Indicates the elevation for a TANK upon a return from a line of sight call.
ZS.LS	A computational variable indicating the Z coordinate of a point S used in Routine LOS.
ZV.LS	A computational variable indicating the Z coordiante at a point V. Used in Routine LOS.
ZW.LS	A computational variable indicating the Z coordinate of a point W. Used in Routine LOS.

ROUTINES AND EVENTS

The following brief descriptions and definitions are useful in locating specific routines and events by their function.

ACTION	A routine used to initiate the movement of appropriate units in conjunction with the leave logic.
AIR.COMMO	A routine used to call for air support for a Blue ground unit by scheduling a B.LUAIRCALL.
AMMO.CHECK	A routine used to check the ammunition status for XM1s.
ARRAY.COORD.SET	A routine used to read in the values for PHSCORD, COCORD, BNCORD and BN.AIR.PRI.
ASSIGN.ORDERS	A routine used to define the areas to which each company can move.
ATRIT	A routine which assesses the attrition against a target which has been hit. It checks for a catastrophic kill, increments mobility and firepower damage, and checks the FKILL,MKILL and MFKILL levels.
ATTEMPT.HANDOFF	A routine used to determine if a non-platoon leader element can select his platoon leader's target.
ATTRITION.CHECK	An event which is used to stop the simulation if the remaining force level of either side becomes too small.
ATT.LIST	A routine used to print the values of the attributes whenever it is called.
BASIC.LOAD	A routine used to initialize the ammunition basic load levels for all elements.
BEST.POS	A routine used to select the best (first) available position number for a vehicle to occupy in a platoon position area.
BL.CREATE	A routine used to create the BLUE (defensive) elements.

B.LUAIRCALL	An event used to file requests for Blue air support in the set WANT.AIR.
BL.FORCES	Used to schedule the creation of BLUE forces.
BN.GO	A routine used to initiate the withdrawal of a BLUE battalion from its current position.
BRST.FIRE	A routine used to determine if small arms direct fire weapons and burst fire dismounted weapons projectiles have impacted on the target.
BSET	A routine used to reset the sector of search and primary direction of heading at the termination of an engagement.
BTLY.CHK	A routine used to increment the casualty counters for monitored weapons systems for use in the leave logic. The argument of this routine is a defender unit.
BUG.CHK	A routine used to determine if the leave logic is to be invoked due to range considerations.
CALL.TO.REMOUNT	A routine used to set up a call to remount dismounted forces on their respective vehicles.
CARDID	A routine used to calculate the time to detection where the target is a vehicle.
CHARGE	An event used to determine if a unit in the hasty defense is to resume the attack.
CHG.SEC.SEARCH	A routine used to modify the search sector of an element.
CO.GO	A routine used to initiate the withdrawal of a BLUE company from its current position.
COMMO.PASS.TGT	A routine used to determine the pointer value to a platoon leader's target.
COMP.SEARCH	A routine used to determine whether any other element in the company is engaging A's potential target S.
COMPUTE	A routine used to determine the conditions under which the engagement occurred.(i.e. target moving, stationary, type of weapon,

PROJO, and range) This routine looks up the standard deviations and biases associated with the assumed normal point of impact.

COORD.SET	A routine used to determine which battalions are to occupy the current coordination line.
CTR.ATK	A routine used to determine if it is appropriate to conduct a limited counterattack based on user input. (CA.DATA)
DANGER.STATE	A routine used to input the TARGET.SELECTION array.
DECISION	A routine used to update the values of CREQST, BNCUR, and BDECUR. This routine is called by Routine ACTION or Routine BUG.CHK to decide if a unit can perform a desired action. It causes entire companies in a battalion to move or all battalions in the brigade set to move if the upper bounds on BNCUR/BDECUR are exceeded.
DECREMENT.AMMO	A routine used to decrement the ammunition level of a specified type.
DEFEND	A routine used to set the positions in a new area attained by the unit, including elements to dismount.
DETECT	An event used to determine whether a detection occurs and to update the detected lists.
DF.CHG	An event used to change the defilade code (DEFNUM) of an element.
DISMTD.CARDIO	A routine used to calculate the time to detection where the target is a dismounted element.
DISMOUNT	A routine used to dismount the ground elements from their fighting vehicles.
DIST	A routine used to calculate the distance between two elements.
DRAW.SABRES	A routine used to initiate counter attacks for the defending unit.
DUMPS	An event used to create punched card

output of the X and Y coordinates of each element for the purpose of plotting.

ELEV	A routine used to compute the macro-terrain elevation for any (x,y) coordinate on the battlefield.
ELEVG	A routine used to provide the macro-terrain elevation for any (x,y) coordinate on the battlefield. Additionally, it computes the gradient.
FINAL.DEATH	An event used to remove MFKILLed TANKs from the battle.
FIRE	An event used to fire a direct fire weapon.
FIRE.SCHEDULE	A routine used to schedule a FIRE event for an element against an opposing element.
FIRST	A routine used to determine the mean of the normal distribution associated with the log-normal distribution of lay times from the array MU.
F.PRINT	A routine used to tally the reasons for aborted fires.
GET	A routine used to access the target selection array pointer, tactics number, acquisition range, range of selected ammo, muzzle of the ammunition, or the fire on the move capability of the firer.
GEOM	A routine used to determine whether a direct fire shot hit the target and to find the correct damage functions and the probability of kill if the shot did hit the target. This is used for vehicle weapons systems.
GET.SP	A routine used to access suppression data.
GET.UP	A routine used to cause a unit in the hasty defense to continue the attack.
HASDEF	A routine used to place attacking elements in a hasty defense based on the attrition level.
HIDE	An event used to place elements in appropriate

defilade conditions.

HIDER	A routine used to determine the micro-terrain elevation for a selected element.
IMPACT	An event used to record the fact that a direct fire round from a firer passed through the target plane.
INF.ARRIVAL	A routine used to determine the appropriate action for a dismounted element to take when he reaches his foxhole after a dismount or upon reaching his vehicle upon a call to remount.
INF.DEST	A routine used to determine the destination for a ground element upon a dismount from his vehicle or upon a call to remount his vehicle.
INIT.POS	A routine used to select the initial position for a vehicle from the position array.
KOVER	A routine used to compute the percent visible when an obstacle is between the observer and the target.
LAY.LOAD	A routine used to compute the lay and load times for a direct fire weapon.
LIMICON	A routine used to determine the probability that an impacting direct fire round is observed by the target based on the target's search sector and primary search direction.
LIST.RIDERS	A routine used to output the names of the riders currently mounted on all of the vehicles in the simulation.
LIST.UPDATE	A routine used to add or delete a target element from a firer's detected list.
LOAD.PLAN	An event used to simulate the loading of pallets of ammunition on supply vehicles at the ATP. An amount of time equal to LOADUP is assessed.
LOC	A routine used to determine whether movement is possible and to in'tiate a call to move.
LOC.UPDATE	An event used to schedule a call to LOC for

position.

PLT.SEARCH	A routine used to determine whether any other element in the platoon is engaging an element's target.
POP.A.MINE	A routine used to assess the results of a mine detonation.
PRIORITY.AND ROUND.SELECT	A routine used to determine the priority of a currently considered target and to select the ammunition to use against that target.
PROXIMITY.DETECT	A routine used to compute detections by the firer of elements which are within close proximity of a target element.
PURGE.LIST	A routine used to purge the target list of a firer for dead targets and for targets to which LOS no longer exists.
REDACT	A routine used to initiate movement actions for attacking elements.
RED.CREATE	A routine used to create RED (attacker) elements.
RELOAD	A routine used to specify the redistribution of ammunition for the XM1.
REMOUNT	A routine used to remount the dismounted elements on their designated vehicles.
RESET	A routine which resets the values of the suppression related attributes of the PLATOON.LEADER permanent entity.
RES1	A routine used to reserve the array dimensions for the accuracy and lethality arrays for dismounted weapons systems.
RES2	A routine used to reserve the array dimensions for accuracy and lethality arrays.
RES3	A routine used to read in specified lethality arrays.
RES4	A routine used to read in specified accuracy arrays.

RES5	A routine used to read in specified accuracy and lethality values for SOVMG,ADDON,DGNV and BUSHBMP.
RES.MOV	A routine used to read in data arrays for the movement routine.
RES.TERR	A routine used to read in data for battlefield definition, terrain hills and forest ellipses.
RT.SEL	A routine used to select the route to be used for a given movement and sets the ROUTE, NEXT.MCP and DIR.ON.RT attributes of the vehicle to correspond to this route.
RTLY.CHK	A routine used to increment the casualty counter for monitored weapon types for use in the leave logic.
SEC.GO	A routine used to move a section of a specified weapon system type within a designated company.
SECTOR.CHECK	A routine used to determine if an opposing force element is in an element's sector.
SET.AREAS	A routine used to set the AREA.START and AREA.END attributes for supply vehicles.
SET.CA	A routine used to read in the counterattack data.
SET.MOVE.AREAS	A routine used to determine the AREA.START and AREA.END attributes for an element.
SET.SP	A routine used to read in the suppression data and to set the suppression levels.
SIGHT	A routine used to set up a line of sight call between two opposing elements.
SNAP.R	A routine used as a debug tool to list attributes.
SORT	A routine used to compute monitored system/weapon attributes.
STAT.CHK	A routine used to calculate the percent of attrition and to determine the appropriate

maneuver action.

STEP.TIME	An event used to schedule detections for all elements currently alive in the simulation.
STOP.SIMULATION	An event used to print the final battle results and to stop the simulation if the user specified time for the run has expired.
STOP.DISMOUNT	A routine used to stop or to restart vehicles before and after a dismount.
STOP.TO.FIRE	A routine used to stop or to restart a vehicle not allowed to fire on the move.
SUBCAL	A routine used to assess the result of an engagement of a mounted weapons system using burst fire of other than the main armament (e.g. coaxial machinegun)
TACTICS	A routine used to select a tactic using a user specified tactic sequence.
TALLY.HIT.STATE	A routine used to record the type of damage sustained by an element, update appropriate sets and to initiate the leave logic.
TARGET.SELECT	An event used to select a target and and schedule a FIRE event based on firer target filters and user specified tactics.
TIM.SP	A routine used to calculate the time delay based on the user supplied suppression data, and the suppression related attributes of the permanent entity PLATOON.LEADER.
TREE.CHECK	A routine used to test the line of sight at the edge of a forest ellipse.
UPLOAD	An event used to "top off" a ground force element with the required ammunition if it is available. This action occurs when the the ground force element reaches its new location in an alternate position.
VEH.GO	A routine used to activate the leave logic for a specified system and weapon code.

elements which are alive.

LOS	A routine used to compute the fraction of the target visible to the observer.
MAX.WAIT.TIME	An event used to initiate the actual movement of elements after a user input delay time subsequent to a call to move.
MOVE	A routine used to update the location, direction and speed of a vehicle to the current simulation time.
MOVE.LIMITS	A routine used to determine the limits on and acceleration with which the vehicle can move.
MOVE.OUT	An event used to initiate movement from ATPs to AMMO.PILEs and the reverse.
NEW.FORCES	An event used to schedule the creation of RED (attacker) forces.
NEWTON	A routine used to compute the tangency point of the lowest sight line over a hill.
OFFLOAD	An event used to remove pallets of ammunition from a supply vehicle at the desired AMMO.PILE.
OTHERGO	A routine used to move all systems other than those designated as a monitored system in the company.
PHAZ.CHK	A routine used to determine if the current coordination line is still occupied. If it is still occupied, the routine reschedules itself, if not it removes the battalion commander from the brigade set and calls COORD.SET.
PILE.SO.CREAT	A routine used to read the resupply input and to print the initial attribute values for each AMMO.PILE and SUPPLY.OFFICER as well as the value of other resupply variables.
PLT.GO	A routine used to initiate the withdrawal of a BLUE (defensive) platoon from its current

WE.HIT	A routine used to schedule an action for a firer following a KKILL on his target.
WE.MISS	A routine used to schedule an action for a firer following an impact which did not result in a KKILL.
WGT.SP	A routine which calculates the weight of a round (relative suppressive effect) and adds that weight to the attribute ROHAT of the appropriate PLATOON.LEADER.
WHERE.THE.HELL.AM.I	A routine used to set the ammunition supply trucks travelling on their routes in the proper direction and in the proper status.
WIDRAW	A routine used to cause units in the hasty defense to withdraw based on attrition level.
XYDUMP	A routine used to punch position and status information out on cards.

BIBLIOGRAPHY

1. Wallace, W.S. and Hagewood, E.G., Simulation of Tactical Alternative Responses (STAR), M.S. Thesis, Naval Postgraduate School, Monterey, CA, December 1978.

2. Caldwell, W.J. and Meiers, W.D., An Air to Ground and Ground to Air Combined Arms Combat Simulation (STAR-AIR), M.S. Thesis, Naval Postgraduate School Monterey, CA, September 1979.

3. Department of the Army, FM 7-20, The Infantry Battalion (Infantry, Airborne, Air Assault, and Ranger), 3 April 1978.

4. Department of the Army, FM 23-3, Tactics, Techniques and Concepts of Antiarmor Warfare, August 1972.

5. Department of the Army, TC 7-24, Antiarmor Tactics and Techniques for Mechanized Infantry, 30 September 1975.

6. Department of the Army, TC 23-23, TOW Heavy Antitank Weapon System, July 1970.

7. Department of the Army, Review of Selected Army Models, May 1971.

8. Defense Intelligence Agency, DDI-1100-77-76, The Soviet Motorized Rifle Company, Octover 1976.

9. Stockfisch, J.A., Models, Data, and War: A Critique of the Study of Conventional Forces, RAND, March 1975.

10. Kiviat, P.J., Villanueva, R., and Markowitz, H.M., SIMSCRIPT II.5 Programming Language, 2d Edition, Consolidated Analysis Center, Inc., 1973.

11. USACDC Report, Army Small Arms Requirements Study (ASARS) Battle Model, Volumes I, II, III, 1972.

12. General Research Corporation, OAD-CR-73, CARMONETTE, Volumes I, II, and III, November 1974.

13. Parry, S.H., and Kelleher, E.P., Tactical Parameters and Input Requirements for the Ground Component of the STAR Combat Model, Naval Postgraduate School Technical Report NPS55-79-023, Monterey, CA, October 1979.

14. Hartman, J.K., Ground Movement Modelling in the STAR Combat Model, Naval Postgraduate School, Technical Report NPS55-80-021, Monterey, CA, May 1980.

15. Hartman, J.K., Parametric Terrain and Line of Sight Modelling in the STAR Combat Model, Naval Postgraduate School, Technical Report NPS55-79-018, August 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4.	Professor James K. Hartman Code 55Hh Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
5.	Professor S. H. Parry, Code 55Py Department of Operations Research Naval Postgraduate School Monterey, California 93940	10
6.	LTC Edward P. Kelleher, Code 55Ka Department of Operations Research Naval Postgraduate School Monterey, California 93940	10
7.	Professor Arthur L. Schoenstadt, Code 53Zh Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
8.	Office of the Commanding General U.S. Army TRADOC ATTN: General Donn A. Starry Ft. Monroe, Virginia 23651	1
9.	Headquarters U.S. Army TRADOC Attn: ATCG-T (Colonel Ed Scribner) Ft. Monroe, Virginia 23651	1

10. Headquarters 1
U.S. Army TRADOC
Attn: Director, Analysis Directorate
Combat Developments
Ft. Monroe, Virginia 23651

11. Headquarters 1
U.S. Army TRADOC
Attn: Director, Maneuver Directorate
Combat Developments (COL Fred Franks)
Ft. Monroe, Virginia 23651

12. Mr. David Hardison 1
Deputy Under Secretary of the Army
(Operations Research)
Department of the Army, The Pentagon
Washington, D.C. 20310

13. LTG William Richardson 1
Commanding General
U.S. Army Combined Arms Center
Ft. Leavenworth, Kansas 66027

14. Director 1
Combined Arms Combat Development Activity
Attn: COL Reed
Ft. Leavenworth, Kansas 66027

15. Director, BSSD
Combined Arms Training Development Activity 1
Attn: ATZLCA-DS
Fort Leavenworth, Kansas 66027

16. Director 1
Combat Analysis Office
Attn: Mr. Kent Pickett
U.S. Army Combined Arms Center
Fort Leavenworth, Kansas 66027

17. Command and General Staff College 1
Attn: Education Advisor
Room 123, Bell Hall
Fort Leavenworth, Kansas 66027

18. Dr Wilbur Payne 1
Director
U.S. Army TRADOC Systems Analysis Activity
White Sands Missile Range, New Mexico 88002

19. Headquarters 1
Department of the Army
Office of the Deputy Chief of Staff
for Operations and Plans
Attn: LTG Glenn Otis
Washington, D.C. 20310

20. Commander 1
U.S. Army Concepts Analysis Agency
8120 Woodmont Avenue
Attn: MOCA-SMS(CPT Steve Shupack)
Bethesda, Maryland 20014

21. Commander 1
U.S. Army Concepts Analysis Agency
Attn: LTC Earl Darden-MOCA-WG
8120 Woodmont Avenue
Bethesda, Maryland 20014

22. Director 1
U.S. Army Night Vision & Electro-optical Lab.
Attn: DEL-NV-VI (Mr. Bob Hermes)
Fort Belvoir, VA 22060

23. Director 1
U.S. Army Material Systems Analysis Activity
Attn: Mr. Will Brooks
Aberdeen Proving Grounds, Maryland 21005

24. Director 1
Armored Combat Vehicle Technology Program
Attn: COL Fitzmorris
U.S. Army Armor Center
Fort Knox, Kentucky 40121

25. Colonel Frank Day 1
TRADOC Systems Manager-XM1
U.S. Army Armor Center
Fort Knox, Kentucky 40121

26. Director 1
Combat Developments, Studies Division
Attn: MAJ W. Scott Wallace
U.S. Army Armor Agency
Fort Knox, KY 40121

27. Commandant 1
U.S. Army Field Artillery School
Attn: ATSF-MBT (CPT Steve Starner)
Fort Sill, Oklahoma 73503

28. Director 1
Combat Developments
Attn: COL Clark Burnett
U.S. Army Aviation Agency
Fort Rucker, Alabama 36360

29. Director 1
Combat Developments
U.S. Army Infantry Agency
Fort Benning, GA 31905

30. Director 1
Missile Intelligence Agency
Attn: ADA Tactics (CPT E.G. Hagewood)
Redstone Arsenal, AL 35809

31. Director 1
Combat Developments
Attn: CPT William D. Meiers
U.S. Army Air Defense Agency
Fort Bliss, TX 79905

32. Commander 1
U.S. Army Logistics Center
Attn: ATCL-OS-Mr Cammeron/CPT Schuessler
Fort Lee, VA 23801

33. Commander 1
USAMMCS
Attn: ATSK-CD-CS-Mr Lee/Mr Marmon
Redstone Arsenal, AL 35809

34. Commander 1
U.S. Army Combined Arms Center
Attn: ATZL-CA-CAT (R.E. DeKinder, Jr)
Fort Leavenworth, KA 66027

35. Director 1
U.S. Army AMSAA
Attn: DRXSY-AA (Mr Tom Coyle)
Aberdeen Proving Grounds, MD 21005

36. Chief 10
TRADOC Research Element Monterey (TREM)
Naval Postgraduate School
Monterey, CA 93940

37. Office of the Deputy Chief of Staff
for Combat Developments 1
U.S. Army TRADOC
Attn: Major General Carl Vuono
Fort Monroe, Virginia 23651

38. Deputy Commanding General 1
Combined Arms Combat Development Activity
Attn: ATZL-CA-DC (BG(P) Jack Walker)
Fort Leavenworth, KA 66027

39. Commanding General 1
U.S. Army Infantry Center
Attn: Major General David Grange
Fort Benning, GA 31905

40. Director 1
Combat Developments
Attn: COL Pokorney
U.S. Army Field Artillery Center
Fort Sill, Oklahoma 73503

41. Director 1
Combat Developments
Attn: COL Gus Watt
U.S. Army Infantry School
Fort Benning, GA 31905

42. Director 1
USATRASANA
Attn: Mr Ray Heath
White Sands Missile Range, New Mexico 88002

43. Commander 1
United States Army ADMINCEN
Attn: ATZI-CD-AD (MAJ Cochard)
Fort Benjamen Harrison, IN 46216

44. Commandant 2
USAIS
Attn: ATSH-CD-CSO-OR (MS Shirley)
Fort Benning, GA 31905

45. Director 1
Doctrine and Leadership
USAIS
Attn: COL Scott
Fort Benning, GA 31905

46. CPT Edward E. Thurman 1
22615 100th SE
Kent, Washington 98012

47. CPT Howard J. Carpenter 5
Naval Postgraduate School
Code 30 (Operations Research)
Monterey, CA 93940

48. Acting Director 5
CASAA
Attn: ATZL-CAR-MD (CPT Reischl)
Fort Leavenworth, KS 66027

Thesis 188888
C265 Carpenter
c.1
Parametric simulation of infantry tactics and equipment (DISMOUNTED-STAR).